DOCUMENT RESUME

ED 324 214 SE 051 593

AUTHOR Lyons, Jim; Verghese, Manoj
TITLE The Challenge '88 Project: Interfacing of Chemical Instruments to Computers.
INSTITUTION John Abbott Coll., Sainte Anne de Bellevue (Quebec).
PUB DATE Aug 88
NOTE 91p.; For the 1989 version, see SE 051 594.
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052) -- Computer Programs (101)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Chemical Analysis; Chemistry; *Computer Software; Computer Uses in Education; Foreign Countries; Higher Education; Instrumentation; Laboratory Procedures; Microcomputers; Science Activities; Science Instruction; *Spectroscopy
IDENTIFIERS Apple (Computer); BASIC Programing Language; IBM Personal Computer; *Laboratory Interfacing; PASCAL Programing Language

ABSTRACT

The main part of this project involved using a computer, either an Apple or an IBM, as a chart recorder for the infrared (IR) and nuclear magnetic resonance (NMR) spectrophotometers. The computer "reads" these machines and displays spectra on its monitor. The graphs can then be stored for future reference and manipulation. The program to be used for Apple is "Graphics Writer" and those to be used with IBM are "Spectrum" and "Realtime". Included with each is the documentation, listing of the program, and sample spectra. The Apple programs used the BASIC language and the IBM used PASCAL. The difficulties encountered with the Apple programs are discussed. A disk catalog is included for the graphic utilities, and the IR and NMR programs. (KR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED324214

THE CHALLENGE '88 PROJECT: INTERFACING OF CHEMICAL INSTRUMENTS TO COMPUTERS

JIM LYONS
MANOJ VERGHESE
DEPARTMENT OF CHEMISTRY

PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

James M. Lyons

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

CEGEP
JOHN ABBOTT
COLLEGE
P.O. BOX / C.P. 2000,
STE. ANNE DE BELLEVUE, QUEBEC H9X 3L9

SE 051 593

Dépôt Légal - 1er trimestre 1990

Bibliothèque Nationale du Québec

© Copyright John Abbott College

THE CHALLENGE '88 PROJECT

INTERFACING OF CHEMICAL INSTRUMENTS TO COMPUTERS

Jim Lyons
Manoj Verghese

August 1988

JOHN ABBOTT COLLEGE
Ste. Anne de Bellevue, Quebec

Table of Contents

The project 3

Apple

Graphics Writer

Documentation 6

Listing 10

Ir.plot

Documentation 21

Listing 25

Sample spectra 31

Nmr.plot

Documentation 35

Listing 39

Sample spectra 47

Nmr.plot.dhgr

Documentation 51

Listing 55

Sample spectra 62

IBM

Spectrum

Documentation 67

Listing 70

Realtime

Documentation 77

Listing 79

Difficulties 82

Disk Catalogs 86

The Project

I was first exposed to computers when I entered high school in 1983. A short time later I aquired my own computer, an Apple IIe. I have been programming ever since. At John Abbott I went on from Basic to learn Fortran and Pascal. But I was to learn more than just programming.

Although I had worked with different languages in various environments, they all shared a common factor. While the programs solved many problems and executed different tasks, they were entirely self-contained. The programs ran happily by themselves with no real consequence. As far as I was concerned they had no practical use. I had heard of the many applications of computers, but never had the opportunity to do any such work with computers myself. This is where Challenge '88 came in.

The main part of the project involved using a computer as a chart recorder for the infrared and nuclear magnetic resonance spectrophotometers. The computer would 'read' these machines and display spectra on its monitor. The graphs could then be stored for future reference and manipulation. With this experience, I have finally seen a useful application of computers in the laboratory.

In conclusion, I must say that this was a summer job with a difference. Instead of working at a garage station or mowing lawns, the experience I gained was very much worthwhile. This is especially true since I will be attending McGill University this fall in their electrical engineering program.

APPLE

GRAPHICS WRITER

Graphics Writer

Introduction

"Graphics Writer" allows you to add titles, labels, remarks etc. to your high resolution graphics pictures. The graphs or pictures are simply bloaded into memory where you can type directly on to the screen. This program was developped by expanding on Beagle Brothers "Hi-Writer" program found on their "Apple Mechanics" diskette.

Requirements

"Graphics Writer" will run on any Apple II series computer with a minimum of 48 k memory. If hardcopies of pictures are desired a graphics supporting printer and an interface card in slot#1 are needed.

Four other files are needed on the disk to run "Graphics Writer"; the "Startup" program, the "Title" binary picture and the two font files, "Cursors" and "Standard". "Startup" bloads the font files and then runs "Graphics Writer".

Main Program: 1500 - 1840

At startup the title and programmer's name are displayed on hi-res page one. The variables are set to their initial values. Page one is then cleared to all black. The program waits for the user to issue a command and then executes it. This cycle repeats until the user quits the program.

Program Title: 2000 - 2170

The trace feature is turned off, the speed set to 255, the text display to normal, the text screen cleared, the hi-res page one turned on, the full screen poked and the color is white.

The graphics file "Title" is bloaded to page one showing the program title and programmer's name with credit to Beagle Brothers.

Initialization of Variables: 3000 - 3360

The program variables are set to their initial values. Definitions can be found at lines 1150 - 1320 in the "Dictionary of Variables" section.

Horizontal Position: 4000 - 4190

Ctrl - X: The text mode is restored and the cursor is erased from page one of hi-res. The user is asked to enter the horizontal position of the cursor. The accepted range is from 1 to 272 inclusive. The answer is error checked to that effect. Page one is redisplayed and the new cursor position is drawn.

Vertical Position: 4200 - 4390

Ctrl - Y: The text mode is restored and the cursor is erased from page one of hi-res. The user is asked to enter the vertical position of the cursor. The accepted range is from 1 to 183 inclusive. The answer is error checked to that effect. Page one is redisplayed and the new cursor position is drawn.

Files: 4400 - 4550

Ctrl - F: The text mode is restored and the current drive is catalogued. The user is asked to press the spacebar to return to page one. Actually, any key will do.

Save Picture: 4600 - 4770

Ctrl - S: The text mode is restored and the user is asked to supply a file name. If no name is entered the user is prompted again. Invalid file names are not checked for (eg. one starting with a number). Page one is then saved to disk. The display is returned to page one with the picture left intact. The user is responsible for ensuring that there is sufficient disk space and that a locked file of the same name does not exist on the disk.

Load Picture: 4800 - 4990

Ctrl - L: The text mode is restored and the user is asked to supply a file name. If no name is entered the user is prompted again. Invalid file names are not checked for (eg. one starting with a number). The file is loaded onto page one and is displayed. The user is responsible for ensuring that the file does exist and that it is a binary file.

Rotational Factor: 5000 - 5150

Ctrl - R: The text mode is restored and the user is asked the enter the rotational factor. The accepted values are 0, 1, 2 and 3, respectively 0, 90, 180 and 270 rotations. When writing vertically (1 or 3), the spacing is not proportional as it is when writing horizontally (0 or 2). The rotational factor stays in effect until a new one is specified.

Clear Screen: 5200 - 5300

Ctrl - C: The color is set to black and a point is plotted. The call 62454 clears the hi-res page (one) to black. The color is restored to white and the cursor is drawn.

Erase Character: 5400 - 5530

Ctrl - E: The color is set to black. A box 7x8 pixels is drawn, with the top corner being the left edge of the cursor. The color is restored to white and the cursor is redrawn since it was erased by the box.

Display Variables: 5600 - 5790

Ctrl - D: The text mode is restored. The horizontal, vertical positions, rotational factor and current picture name are shown. The user presses the spacebar to return to page one. Actually, any key will do.

Print Screen: 5800 - 5920

Ctrl - P: The picture in page one of hi-res is dumped to the printer. The interface card is assumed to be a Grappler+ installed in slot #1. Output is restored to the monitor with the 80 column card on in 40 column mode. Display is returned to page one.

Quit: 6000 - 6050

Ctrl - Q: The text screen is cleared and then revealed. The "Graphics Writer" program is cleared from memory, but page one is left intact.

Cursor Increment: 6200 - 6270

Ctrl - T: The horizontal cursor increment is toggled between 1 and 7, the vertical increment between 1 and 8 pixels.

Cursor Movement: 6400 - 6540

Arrow keys: The cursor is erased. The new position is calculated according to which arrow key was pressed. A check is performed to keep the cursor within the bounds of the screen. The cursor is drawn at the new position. The cursor will 'wrap around' if it moves off an edge of the screen.

Erase Cursor: 6600 - 6670

The color is set to black, the cursor erased and the color restored to white.

Draw Cursor: 6800 - 6860

The color is set to white and the cursor is drawn.

Cursor Position Validity: 7000 - 7080

The horizontal position is kept between 1 and 272 inclusively, the vertical between 1 and 183 inclusively. Positions which are out of bounds are 'wrapped around'.

Print Characters - Beagle Bros "Hi - Writer": 7200 - 7380

Characters: The cursor is erased. The Beagle Bros routine from "Hi-Writer" is used to draw the characters on page one. The new cursor position is determined and then drawn.

Help Screen: 7400 - 7590

Ctrl - I: The text mode is restored and put into 80 column mode. The fourteen commands along with explanations are displayed one at a time. The user presses the space bar to step through them. When done, the text screen is put back to 40 columns and the display returns to page one.

```
]LIST

100  REM  ..........................................................
110 :
120  REM  Date      :  June 9, 1988
130  REM  Author    :  Manoj Verghese
140 :
150  REM  ........................Load Fonts ......................
160 :
170 D$ =  CHR$ (13) +  CHR$ (4)
180 :
190  PRINT D$;"BLOAD CURSORS,A16384"
200  PRINT D$;"BLOAD STANDARD,A16640"
210 :
220  POKE 800,0
230  POKE 801,65
240 :
250  REM  ........................... Writer ..........................
260 :
270  PRINT  CHR$ (4);"RUN GRAPHICS WRITER"

]
]LIST

10 ABOVE = 24576 + 1
20  IF  PEEK (103) +  PEEK (104) * 256 = ABOVE THEN 1000
30  POKE ABOVE - 1,0
40  POKE 103,ABOVE -  INT (ABOVE / 256) * 256
50  POKE 104, INT (ABOVE / 256)
60  PRINT  CHR$ (4);"RUN GRAPHICS WRITER"
70 :
1000  REM  ..........................................................
1010 :
1020  REM  A little explanation is in order for the above.
1030 :
1040  REM  A program longer than $1800 bytes that uses hi-res graphics
1050  REM  must be loaded starting at $6000.  If loaded at $800, memory
1060  REM  not be sufficient.  The lines above from Beagle Bros do this.
1070 :
1080  REM  ..........................................................
1090 :
1100  REM  Date    :  June 6, 1988
1110  REM  Program :  Mixed Graphics & Text
1120  REM  Authors :  Manoj Verghese
1130  REM          :  Beagle Brothers - " Hi - Writer "
1140 :
1150  REM  ................... Dictionary of Variables ..................
1160 :
1170  REM  Above      :  Memory start location for program
1180  REM  Do$        :  User command
1190  REM  Ht,Ht$     :  Horiztonal position of cursor
1200  REM  Vt,Vt$     :  Vertical position
1210  REM  Rot,Rt,    :  Rotational factor
1220  REM  Scale      :  Scale factor
1230  REM  Hires      :  poke for hires
1240  REM  Page1      :  poke for page1
1250  REM  L1,L2,Loc  :  font locations
1260  REM  D$         :  chr$(13) + chr$(4) - to load fonts
1270  REM  Xinc,Yinc  :  increments for cursor movement
```

```
1280  REM  Com$(14)   :  list of commands in help screen
1290  REM  Hit$       :  hit spacebar to continue ( get )
1300  REM  Pict$      :  name of picture
1310  REM  i          :  loop counter
1320 :
1500  REM  ................. Main Program Execution ....................
1510 :
1520  REM  Display Program Title
1530  GOSUB 2000
1540 :
1550  REM  Initialize Variables
1560  GOSUB 3000
1570 :
1580  REM  Set Hi-res graphics screen
1590  GOSUB 5200
1600 :
1610  REM  Get User Command
1620  GET DO$
1630 :
1640  IF DO$ =  CHR$ (24) THEN  GOSUB 4000: REM  ctrl - X ' x axis
1650  IF DO$ =  CHR$ (25) THEN  GOSUB 4200: REM  ctrl - Y ' y axis
1660  IF DO$ =  CHR$ (06) THEN  GOSUB 4400: REM  ctrl - F ' files
1670  IF DO$ =  CHR$ (19) THEN  GOSUB 4600: REM  ctrl - S ' save
1680  IF DO$ =  CHR$ (12) THEN  GOSUB 4800: REM  ctrl - L ' load
1690  IF DO$ =  CHR$ (18) THEN  GOSUB 5000: REM  ctrl - R ' rotate
1700  IF DO$ =  CHR$ (03) THEN  GOSUB 5200: REM  ctrl - C ' clear
1710  IF DO$ =  CHR$ (05) THEN  GOSUB 5400: REM  ctrl - E ' erase
1720  IF DO$ =  CHR$ (04) THEN  GOSUB 5600: REM  ctrl - D ' display
1730  IF DO$ =  CHR$ (16) THEN  GOSUB 5800: REM  ctrl - P ' print
1740  IF DO$ =  CHR$ (17) THEN  GOSUB 6000: REM  ctrl - Q ' quit
1750  IF DO$ =  CHR$ (20) THEN  GOSUB 6200: REM  ctrl - T ' toggle
1760  IF DO$ =  CHR$ (08) THEN  GOSUB 6400: REM        <- ' left
1770  IF DO$ =  CHR$ (32) THEN  GOSUB 6400: REM        -> ' right
1780  IF DO$ =  CHR$ (10) THEN  GOSUB 6400: REM       ->~ ' down
1790  IF DO$ =  CHR$ (11) THEN  GOSUB 6400: REM       ~<- ' up
1800  IF DO$ =  CHR$ (09) THEN  GOSUB 7400: REM  ctrl - I ' information
1810  IF  ASC (DO$) > 32 AND  ASC (DO$) < 128 THEN  GOSUB 7200
1820 :
1830  GOTO 1620
1840 :
2000  REM  ....................... Program Title .....................
2010 :
2020  NOTRACE
2030  SPEED= 255
2040  NORMAL
2050  HOME
2060  TEXT
2070 :
2080  HGR
2090  POKE 49234,0
2100  HCOLOR= 3
2110 :
2120  PRINT  CHR$ (4);"BLOAD TITLE,A$2000"
2130 :
2140  FOR I = 1 TO 1000: NEXT I
2150 :
2160  RETURN
2170 :
3000  REM  ................ Initialization of Variables ..............
3010 :
```

```
3020 HT = 1
3030 VT = 1
3040 RT = 0
3050  SCALE= 1
3060 :
3070 HIRES = 49232
3080 PAGE1 = 49236
3090 :
3100 L1 =  PEEK (800)
3110 L2 =  PEEK (801)
3120 LOC =  PEEK (800) +  PEEK (801) * 256
3130 :
3140 D$ =  CHR$ (13) +  CHR$ (4)
3150 :
3160 XINC = 7
3170 YINC = 8
3180 :
3190  DIM COM$(14)
3200 COM$(01) = "ctrl - x"
3210 COM$(02) = "ctrl - y"
3220 COM$(03) = "ctrl - f"
3230 COM$(04) = "ctrl - s"
3240 COM$(05) = "ctrl - l"
3250 COM$(06) = "ctrl - r"
3260 COM$(07) = "ctrl - c"
3270 COM$(08) = "ctrl - e"
3280 COM$(09) = "ctrl - d"
3290 COM$(10) = "ctrl - p"
3300 COM$(11) = "ctrl - q"
3310 COM$(12) = "ctrl - t"
3320 COM$(13) = "arrow keys"
3330 COM$(14) = "ctrl - i"
3340 :
3350  RETURN
3360 :
4000  REM  ..................... Horizontal Position .....................
4010 :
4020  TEXT : HOME
4030 :
4040  GOSUB 6600: REM  Erase cursor
4050 :
4060  VTAB 10
4070  PRINT "     Enter the horizontal position": PRINT
4080  INPUT "     of the cursor ( 1 - 272 )    : ";HT$
4090 :
4100 HT =  ABS ( INT ( VAL (HT$)))
4110  IF HT < 1 OR HT > 272 THEN 4060
4120 :
4130  POKE HIRES,0
4140  POKE PAGE1,0
4150 :
4160  GOSUB 6800: REM  Draw cursor
4170 :
4180  RETURN
4190 :
4200  REM  ..................... Vertical Position .....................
4210 :
4220  TEXT : HOME
4230 :
4240  GOSUB 6600: REM  Erase cursor
```

```
4250 :
4260  VTAB 10
4270  PRINT "      Enter the vertical position": PRINT
4280  INPUT "      of the cursor ( 1 - 183 ) : ";VT$
4290 :
4300 VT =  ABS ( INT ( VAL (VT$)))
4310  IF VT < 1 OR VT > 184 THEN 4260
4320 :
4330  POKE HIRES,0
4340  POKE PAGE1,0
4350 :
4360  GOSUB 6800: REM  Draw cursor
4370 :
4380  RETURN
4390 :
4400  REM  ........................... Files ............................
4410 :
4420  TEXT
4430  HOME
4440 :
4450  PRINT
4460  PRINT  CHR$ (4);"CATALOG"
4470  PRINT
4480  PRINT
4490  HTAB 12: PRINT "Hit the spacebar ";: GET HIT$
4500 :
4510  POKE HIRES,0
4520  POKE PAGE1,0
4530 :
4540  RETURN
4550 :
4600  REM  ......................... Save Picture ........................
4610 :
4620  TEXT
4630  HOME
4640 :
4650  VTAB 10
4660  INPUT "   Name of Picture to save : ";PICT$
4670  IF PICT$ = "" THEN 4730
4680 :
4690  PRINT
4700  PRINT  CHR$ (4);"BSAVE ";PICT$;",A$2000,L$1FFF"
4710  PRINT
4720 :
4730  POKE HIRES,0
4740  POKE PAGE1,0
4750 :
4760  RETURN
4770 :
4800  REM  ......................... Load Picture ........................
4810 :
4820  TEXT
4830  HOME
4840 :
4850  VTAB 10
4860  INPUT "   Name of Picture to load : ";PICT$
4870  IF PICT$ = "" THEN 4930
4880 :
4890  PRINT
4900  PRINT  CHR$ (4);"BLOAD ";PICT$;",A$2000"
```

```
4910  PRINT
4920 :
4930  POKE HIRES,0
4940  POKE PAGE1,0
4950 :
4960  GOSUB 6800: REM  Draw Cursor
4970 :
4980  RETURN
4990 :
5000  REM  ..................... Rotational Factor .....................
5010 :
5020  TEXT
5030  HOME
5040 :
5050  VTAB 10
5060  INPUT "       Enter rotational factor : ";RT$
5070 :
5080 RT =  ABS ( INT ( VAL (RT$)))
5090  IF RT < 0 OR RT > 3 THEN 5060
5100 :
5110  POKE HIRES,0
5120  POKE PAGE1,0
5130 :
5140  RETURN
5150 :
5200  REM  ......................... Clear Screen .....................
5210 :
5220  HCOLOR= 0
5230  HPLOT 0,0
5240  CALL 62454
5250  HCOLOR= 3
5260 :
5270  GOSUB 6800: REM  Draw cursor
5280 :
5290  RETURN
5300 :
5400  REM  ......................... Erase Character .....................
5410 :
5420  HCOLOR= 0
5430 :
5440  FOR I = 0 TO 7
5450  HPLOT HT,I + VT TO HT + 6,I + VT
5460  NEXT I
5470 :
5480  HCOLOR= 3
5490 :
5500  GOSUB 6800: REM  Draw cursor
5510 :
5520  RETURN
5530 :
5600  REM  ..................... Display Variables .....................
5610 :
5620  TEXT
5630  HOME
5640 :
5650  VTAB 07: HTAB 06: PRINT "Horizontal Position ( Ht ) : ";HT
5660 :
5670  VTAB 09: HTAB 06: PRINT "Vertical Position   ( Vt ) : ";VT
5680 :
5690  VTAB 11: HTAB 06: PRINT "Rotational Factor   ( Rt ) : ";RT
```

```
5700 :
5710  VTAB 13: HTAB 06: PRINT "Current Picture  ( Pict$ ) : ";PICT$
5720 :
5730  VTAB 17: HTAB 11: PRINT "Hit the spacebar : ";: GET HIT$
5740 :
5750  POKE HIRES,0
5760  POKE PAGE1,0
5770 :
5780  RETURN
5790 :
5800  REM  ..................... Print Screen .....................
5810 :
5820  PRINT
5830  PRINT  CHR$ (4);"PR#1"
5840  PRINT  CHR$ (9);"GDR"
5850  PRINT  CHR$ (4);"PR#3"
5860  PRINT  CHR$ (27); CHR$ (17)
5870 :
5880  POKE HIRES,0
5890  POKE PAGE1,0
5900 :
5910  RETURN
5920 :
6000  REM  ......................... Quit .........................
6010 :
6020  HOME
6030  TEXT
6040  NEW
6050 :
6200  REM  ..................... Cursor Increment .....................
6210 :
6220 XINC = 8 - XINC: REM  Toggles between ( 1, 7 )
6230 :
6240 YINC = 9 - YINC: REM  Toggles between ( 1, 8 )
6250 :
6260  RETURN
6270 :
6400  REM  ..................... Cursor Movement .....................
6410 :
6420  GOSUB 6600: REM  Erase cursor
6430 :
6440  IF DO$ =  CHR$ (08) THEN HT = HT - XINC
6450  IF DO$ =  CHR$ (32) THEN HT = HT + XINC
6460  IF DO$ =  CHR$ (10) THEN VT = VT + YINC
6470  IF DO$ =  CHR$ (11) THEN VT = VT - YINC
6480 :
6490  GOSUB 7000: REM  Check validity of cursor position
6500 :
6510  GOSUB 6800: REM  Draw cursor
6520 :
6530  RETURN
6540 :
6600  REM  ..................... Erase Cursor .....................
6610 :
6620  HCOLOR= 0
6630  HPLOT HT,VT TO HT + 1,VT
6640  HCOLOR= 3
6650 :
6660  RETURN
6670 :
```

```
6800  REM  ....................... Draw Cursor .......................
6810 :
6820  HCOLOR= 3
6830  HPLOT HT,VT TO HT + 1,VT
6840 :
6850  RETURN
6860 :
7000  REM  ................. Cursor Position Validity .................
7010 :
7020  IF HT < 1 THEN HT = HT + 272
7030  IF HT > 272 THEN HT = HT - 272
7040  IF VT < 1 THEN VT = VT + 184
7050  IF VT > 184 THEN VT = VT - 184
7060 :
7070  RETURN
7080 :
7200  REM  .... Print Characters - Beagle Brothers " Hi - Writer " .....
7210 :
7220  GOSUB 6600: REM   Erase cursor
7230 :
7240  POKE 232,L1
7250  POKE 233,L2
7260  ROT= RT * 16
7270  DRAW 100 AT HT,VT: DRAW  ASC (DO$) - 31: XDRAW 99
7280  CALL 62923
7290 HT =  PEEK (224) +  PEEK (225) * 256
7300  IF RT = 1 THEN VT = VT + 8
7310  IF RT = 3 THEN VT = VT - 8
7320 :
7330  GOSUB 7000: REM  Check cursor position validity
7340 :
7350  GOSUB 6800: REM  Draw cursor
7360 :
7370  RETURN
7380 :
7400  REM  ......................... Help Screen .........................
7410 :
7420  HOME : TEXT
7430 :
7440  PRINT : PRINT  CHR$ (4);"PR#3": PRINT
7450 :
7460  FOR I = 1 TO 14
7470 :
7480  GOSUB 7600: REM  Clear text screen, print heading
7490  ON I GOSUB 7700,7800,7900,8000,8100,8200,8300,8400,8500,8600,8700,8
     800,8900,9000
7500  VTAB 24: POKE 1403,31: PRINT "Hit the spacebar ";: GET HIT$
7510 :
7520  NEXT I
7530 :
7540  PRINT  CHR$ (27); CHR$ (17)
7550  POKE HIRES,0
7560  POKE PAGE1,0
7570 :
7580  RETURN
7590 :
7600  REM  ......................... Print Heading .....................
7610 :
7620  HOME
7630  VTAB 2: POKE 1403,30: PRINT "====================="
```

```
7640  VTAB 3: POKE 1403,40 -  LEN (COM$(I)) / 2: PRINT COM$(I)
7650  VTAB 4: POKE 1403,30: PRINT "====================="
7660  VTAB 10
7670 :
7680  RETURN
7690 :
7700  REM  ........................... Ctrl - X ..........................
7710 :
7715  PRINT : POKE 1403,17
7720  PRINT "Ctrl - x  allows you to enter the x co-ordinate"
7725  PRINT : POKE 1403,17
7730  PRINT "of the cursor numerically.  The range is from  "
7735  PRINT : POKE 1403,17
7740  PRINT "1 to 272 ( pixels ) on the hi - res screen.    "
7750 :
7760  RETURN
7770 :
7800  REM  ........................... Ctrl - Y ..........................
7810 :
7815  PRINT : POKE 1403,17
7820  PRINT "Ctrl - y  allows you to enter the y co-ordinate"
7825  PRINT : POKE 1403,17
7830  PRINT "of the cursor numerically.  The range is from  "
7835  PRINT : POKE 1403,17
7840  PRINT "1 to 183 ( pixels ) on the hi - res screen.    "
7850 :
7860  RETURN
7870 :
7900  REM  ........................... Ctrl - F ..........................
7910 :
7915  PRINT : POKE 1403,17
7920  PRINT "Ctrl - F  will catalog the disk.  Pressing the"
7925  PRINT : POKE 1403,17
7930  PRINT "the spacebar will cause the display to return "
7935  PRINT : POKE 1403,17
7940  PRINT "the current picture.                          "
7950 :
7960  RETURN
7970 :
8000  REM  ........................... Ctrl - S ..........................
8010 :
8015  PRINT : POKE 1403,17
8020  PRINT "Ctrl - S  will allow you to save the current"
8025  PRINT : POKE 1403,17
8030  PRINT "graphics picture ( page 1 ) to the disk.  It"
8035  PRINT : POKE 1403,17
8040  PRINT "will ask you to supply a file name.          "
8050 :
8060  RETURN
8070 :
8100  REM  ........................... Ctrl - L ..........................
8110 :
8115  PRINT : POKE 1403,17
8120  PRINT "Ctrl - L  will allow you to load a graphics"
8125  PRINT : POKE 1403,17
8130  PRINT "picture from disk to page 1 of hi -res.  It"
8135  PRINT : POKE 1403,17
8140  PRINT "will of course, prompt you for a file name."
8150 :
8160  RETURN
```

```
8170 :
8200  REM  ........................... Ctrl - R ..........................
8210 :
8215  PRINT : POKE 1403,17
8220  PRINT "Ctrl - r will allow you to specify a rotational"
8225  PRINT : POKE 1403,17
8230  PRINT "factor.  It is restricted to 0, 1, 2, and 3.    "
8235  PRINT : POKE 1403,17
8240  PRINT "These correspond to 0, 90, 180, 270 degrees.    "
8245  PRINT : POKE 1403,17
8250  PRINT "When typing is resumed the characters will be   "
8255  PRINT : POKE 1403,17
8260  PRINT "accordingly rotated.                            "
8270 :
8280  RETURN
8290 :
8300  REM  ........................... Ctrl - C ..........................
8310 :
8315  PRINT : POKE 1403,17
8320  PRINT "Ctrl - c  clears the hi - res graphics ( page 1 )"
8325  PRINT : POKE 1403,17
8330  PRINT "to all black.  If the current picture is not     "
8335  PRINT : POKE 1403,17
8340  PRINT "saved, it will be lost.                          "
8350 :
8360  RETURN
8370 :
8400  REM  ........................... Ctrl - E ..........................
8410 :
8415  PRINT : POKE 1403,17
8420  PRINT "Ctrl - e  erases the area occupied one character."
8425  PRINT : POKE 1403,17
8430  PRINT "Specifically, a box 7 wide, 8 long ( pixels ).   "
8440 :
8450  RETURN
8460 :
8500  REM  ........................... Ctrl - D ..........................
8510 :
8515  PRINT : POKE 1403,17
8520  PRINT "Ctrl - d  displays the relevant program variables"
8525  PRINT : POKE 1403,17
8530  PRINT "and their respective values.                     "
8540 :
8550  RETURN
8560 :
8600  REM  ........................... Ctrl - P ..........................
8610 :
8615  PRINT : POKE 1403,17
8620  PRINT "Ctrl - p  dumps the current hi - res picture "
8625  PRINT : POKE 1403,17
8630  PRINT "( page 1 ) to the printer.  Some alterations "
8635  PRINT : POKE 1403,17
8640  PRINT "may have to be made to lines 5830 - 5850 to  "
8645  PRINT : POKE 1403,17
8650  PRINT "suit your particular system's idiosyncracies."
8660 :
8670  RETURN
8680 :
8700  REM  ........................... Ctrl - Q ..........................
8710 :
```

```
8715  PRINT r POKE 1403,17
8720  PRINT "Ctrl - Q  quits the program, clears it from "
8725  PRINT : POKE 1403,17
8730  PRINT "memory and places the display in text mode. "
8735  PRINT : POKE 1403,17
8740  PRINT "The pictures and bloaded files will however "
8745  PRINT : POKE 1403,17
8750  PRINT "remain in memory.                            "
8760 :
8770  RETURN
8780 :
8800  REM  ........................... Ctrl - T ........................
8810 :
8815  PRINT : POKE 1403,17
8820  PRINT "Ctrl - t  toggles the cursor movement increment."
8825  PRINT : POKE 1403,17
8830  PRINT "The x,y increment is 7,8 or 1,1 respectively.   "
8835  PRINT : POKE 1403,17
8840  PRINT "The former steps the cursor by 'character size'."
8845  PRINT : POKE 1403,17
8850  PRINT "The latter steps the cursor by single pixels.   "
8860 :
8870  RETURN
8880 :
8900  REM  ........................... Arrow Keys ......................
8910 :
8915  PRINT : POKE 1403,17
8920  PRINT "The arrow keys move the cursor on the screen"
8925  PRINT : POKE 1403,17
8930  PRINT "immediately ( as opposed to entering numbers"
8935  PRINT : POKE 1403,17
8940  PRINT "with ctrl - x,y ).                           "
8950 :
8960  RETURN
8970 :
9000  REM  ........................... Ctrl - I ........................
9010 :
9015  PRINT : POKE 1403,17
9020  PRINT "Well, you must know all about ctrl - i otherwise"
9025  PRINT : POKE 1403,17
9030  PRINT "you wouldn't be here.  Unless of course, you hit"
9035  PRINT : POKE 1403,17
9040  PRINT "ctrl - i by mistake.  In any case you know what "
9045  PRINT : POKE 1403,17
9050  PRINT "what it does.  ( watch what you type, it helps )"
9060 :
9070  RETURN
9080 :

]
```

IR.PLOT

Ir.Plot

Introduction

"Ir.plot" is a program which runs with the Perkin Elmer Infrared Spectrophotometer. This program is meant to replace the chart recorder that is found on the Ir. "Ir.plot" not only produces spectra but can save the graphs and data for later reference and manipulation. Futhermore, the "Graphics Writer" program can be used to label the graphs once "Ir.plot" has produced them.

Requirements

"Ir.plot" will run on any Apple II series computer with a minimum of 48 k memory and two disk drives. A twelve bit analog to digital converter is needed to convert the incoming voltage for the Ir. An "Applied Engineering" converter was used for our project. For more information please refer to the "12 bit A/D Users manual". If hardcopies of pictures are desired a graphics supporting printer and an interface card in slot#1 are needed.

One other file is needed on the disk to run "Ir.plot"; the "Axis.ir" binary file. This provides a set of labeled axes for spectrum to be drawn on.

Main Program: 1390 - 1550

At startup the title and programmer's name are displayed on text screen. After a brief pause, the variables are initialized. The menu is then displayed. The program waits for the user to select one of the nine options. After having completed the execution of an option the program returns to the menu.

Initialization of Variables: 1560 - 1870

The program variables are set to their initial values. Please refer to the "Dictionary of Variables" found at lines 1100 to 1380 for the definitions.

1. Instructions: 2000 - 2070

The instruction module for this program has not been written.

2. Data Collection: 3000 - 3430

A message is displayed instructing the user to start the Ir scan. The Ir must be in the fast scan mode which takes approximately two minutes. The chart must be pushed to the extreme right, where the position of the pen is to the left of 4000. The 'return' and 'scan' buttons should be pressed simultaneously for best results. Sampling will begin at 4000 and end at 600.

3. Save Hires 2 to binary: 4000 - 4090

Drive 2 is catalogued and the user is asked for a filename. Hi-res page two is then saved to disk. The user is responsible for ensuring that there is sufficient disk space and a locked file of the same name does not exist.

4. Save data array to text file: 5000 - 5180

Drive 2 is catalogued and the user is asked for a filename. The sequential textfile is opened and the data is recorded. The user is responsible for ensuring that there is sufficient disk space and a locked file of the same name does not exist.

5. Load graph from disk: 6000 - 6150

Drive 2 is catalogued and the user is asked for a filename. The user is told of the viewing instructions. The binary file is loaded on to page two and the display switches to hi-res. The user is responsible for ensuring that the file does exist and is a binary file.

6. Load data from text file: 7000 - 7240

Drive 2 is catalogued and the user is asked for a file name. The user is told of the viewing instructions. The file is verified to determine if it exists. Once the data is read, the

display switches to page two and the graph is drawn. The user is responsible for specifying the correct file name.

7. Print Routine: 8000 - 8160

The user is told to set up the printer and press any key when ready. Hi-res page two is rotated and double sized when dumped. The interface card is assumed to be a Grappler+ installed in slot #1.

8. Turn on hires page 2: 9000 - 9120

Hi-res page two is displayed. When the user presses any key the program returns to the menu.

Title Page: 10 000 - 10 150

The program title and programmers' names are displayed. After a brief pause, the menu is displayed.

Display Menu: 11 000 - 11 230

The menu consisting of nine options is shown. The user selects one of the nine. If the number chosen is invalid the user is prompted again.

Screen Plot: 12 000 - 12 180

The display is switched to hi-res page two. The color is set to white. The predrawn axes "Axis.ir" is loaded onto page two. Since 1116 points have been collected but only 279 are required for one screen, the graph is drawn using every fourth point.

Show stats on data: 13 000 - 13 170

The minimum, maximum values and the range of data is shown. These values are then reset. The user is asked to press any key to continue.

File name: 14 000 - 14 160

Drive 2 is catalogued and the user is asked for a file name. If no name is entered the user is prompted again. Invalid file names are not checked for (eg. one starting with a number).

Viewing Instructions: 15 000 - 15 110

The user is told that the file specified is being loaded. Once the graph is drawn, pressing any key will cause the progam to return to the menu.

```
]LIST

1000  REM  ------------------------------------------------------------
1010  REM  !                  Infrared Spectrum Plotter                 !
1020  REM  ------------------------------------------------------------
1030  :
1040  REM  Date       :    June 14, 1988
1050  REM  Program    :    Irplot.2  ( modified from GCplot.3 )
1060  REM  Authors    :    Jim Lyons       - original version
1070  REM                  Manoj Verghese  - modification for IR
1080  REM                                  - listing format
1090  :
1100  REM  ------------------------------------------------------------
1110  REM  !                    Dictionary of Variables                 !
1120  REM  ------------------------------------------------------------
1130  :
1140  REM  Channel        :   channel of card for data sampling
1150  REM  Gain           :   gain setting for card
1160  REM  Cslot          :   slot number of card
1170  REM  Addr           :   address of card
1180  :
1190  REM  Number         :   number of samples taken
1200  REM  Z(i)           :   array to store samples
1210  REM  Za             :   z(i) given average value of za
1220  :
1230  REM  Over           :   threshold value for sample readings
1240  REM  Max            :   maximum value from samples
1250  REM  Min            :   minimum value from samples
1260  :
1270  REM  D$             :   control - d
1280  REM  I$             :   control - i
1290  :
1300  REM  An$            :   user selected menu option
1310  REM  Key$           :   user hits any key to continue
1320  REM  File$          :   name of graphics or data file
1330  :
1340  REM  i,j            :   loop counters - indices of z
1350  :
1360  REM  x1, y1         :   start point of line being plotted
1370  REM  x2, y2         :   end point of line being plotted
1380  :
1390  REM  ------------------------------------------------------------
1400  REM  !                        Main Execution                      !
1410  REM  ------------------------------------------------------------
1420  :
1430  GOSUB 10000: REM  Title Page
1440  :
1450  GOSUB 1560: REM  Initialize variables
1460  :
1470  GOSUB 11000: REM  Menu
1480  :
1490  ON  VAL (AN$) GOSUB 2000,3000,4000,5000,6000,7000,8000,9000
1500  :
1510  IF AN$ <  > "9" THEN 1470
1520  :
1530  HOME
1540  END
1550  :
1560  REM  ------------------------------------------------------------
```

```
1570  REM  !                    Initialization of variables                !
1580  REM  ---------------------------------------------------------------
1590 :
1600 CHANNEL = 0
1610 GAIN = 3
1620 CSLOT = 7
1630 ADDR = 49280 + CSLOT * 16
1640 :
1650 NUMBER = 1116
1660  DIM Z(NUMBER)
1670 ZA = 0
1680 :
1690 OVER = 4090
1700 MAX = 0
1710 MIN = 4096
1720 :
1730 D$ =  CHR$ (4)
1740 I$ =  CHR$ (9)
1750 :
1760 AN$ = ""
1770 KEY$ = ""
1780 FILE$ = ""
1790 :
1800 I = 0
1810 J = 0
1820 :
1830 X1 = 0:Y1 = 0
1840 X2 = 0:Y2 = 0
1850 :
1860  RETURN
1870 :
2000  REM  ---------------------------------------------------------------
2010  REM  !                    1.  Instructions Routine                   !
2020  REM  ---------------------------------------------------------------
2030 :
2040  REM  Sorry, no instructions ready
2050 :
2060  RETURN
2070 :
3000  REM  ---------------------------------------------------------------
3010  REM  !                      2.  Data Collection                      !
3020  REM  ---------------------------------------------------------------
3030 :
3040  REM  ..................... sampling message .....................
3050 :
3060  HOME
3070 :
3080  VTAB 10: HTAB 08: PRINT "Prepare spectrophotometer"
3090  VTAB 12: HTAB 08: INPUT "When ready press return ";KEY$
3100  VTAB 16: HTAB 12: PRINT "Sampling Data ..."; CHR$ (7)
3110 :
3120  FOR I = 1 TO 2600: NEXT I: REM  synchronize with IR
3130 :
3140  REM  ..................... read a/d card .....................
3150 :
3160  POKE ADDR + 3,GAIN * 16 + CHANNEL: REM  initializes a/d card
3170 :
3180  FOR I = 1 TO NUMBER
3190 ZA = 0
3200 :
```

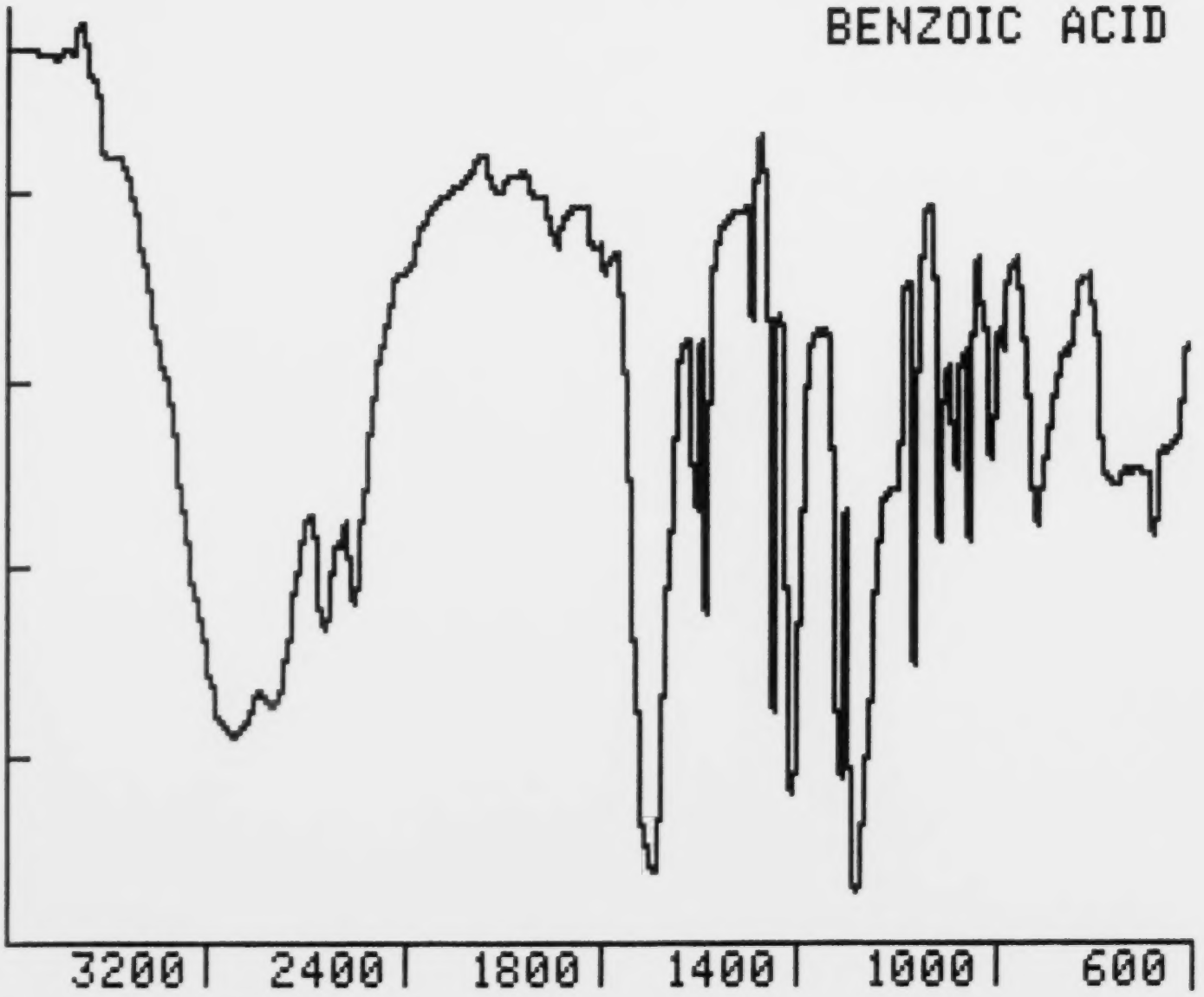
```
3210  FOR J = 1 TO 7
3220  POKE ADDR + 2,0
3230 ZA = ZA +  PEEK (ADDR + 1) * 256 +  PEEK (ADDR)
3240  NEXT J
3250 :
3260 Z(I) =  INT (ZA / 7 + 0.5)
3270  NEXT I
3280 :
3290  REM  ................. analysis and display of data ..............
3300 :
3310  FOR I = 1 TO NUMBER
3320  IF Z(I) > MAX THEN MAX = Z(I)
3330  IF MIN > Z(I) THEN MIN = Z(I)
3340  NEXT I
3350 :
3360  GOSUB 13000: REM  show statistics on data
3370 :
3380  GOSUB 12000: REM  screen plot
3390 :
3400  GET KEY$
3410 :
3420  RETURN
3430 :
4000  REM  ------------------------------------------------------------
4010  REM  |                3.  Save Hires 2 to Binary                |
4020  REM  ------------------------------------------------------------
4030 :
4040  GOSUB 14000: REM  Get file name
4050 :
4060  PRINT D$;"BSAVE ";FILE$;",A$4000,L$1FFF"
4070 :
4080  RETURN
4090 :
5000  REM  ------------------------------------------------------------
5010  REM  |              4.  Save data array to text file             |
5020  REM  ------------------------------------------------------------
5030 :
5040  HOME
5050 :
5060  GOSUB 14000: REM  Get file name
5070 :
5080  PRINT D$;"OPEN ";FILE$
5090  PRINT D$;"WRITE ";FILE$
5100 :
5110  FOR I = 1 TO NUMBER
5120  PRINT Z(I)
5130  NEXT I
5140 :
5150  PRINT D$;"CLOSE ";FILE$
5160 :
5170  RETURN
5180 :
6000  REM  ------------------------------------------------------------
6010  REM  |                  5.  Load graph from disk                 |
6020  REM  ------------------------------------------------------------
6030 :
6040  HOME
6050 :
6060  GOSUB 14000: REM  Get file name
6070 :
```

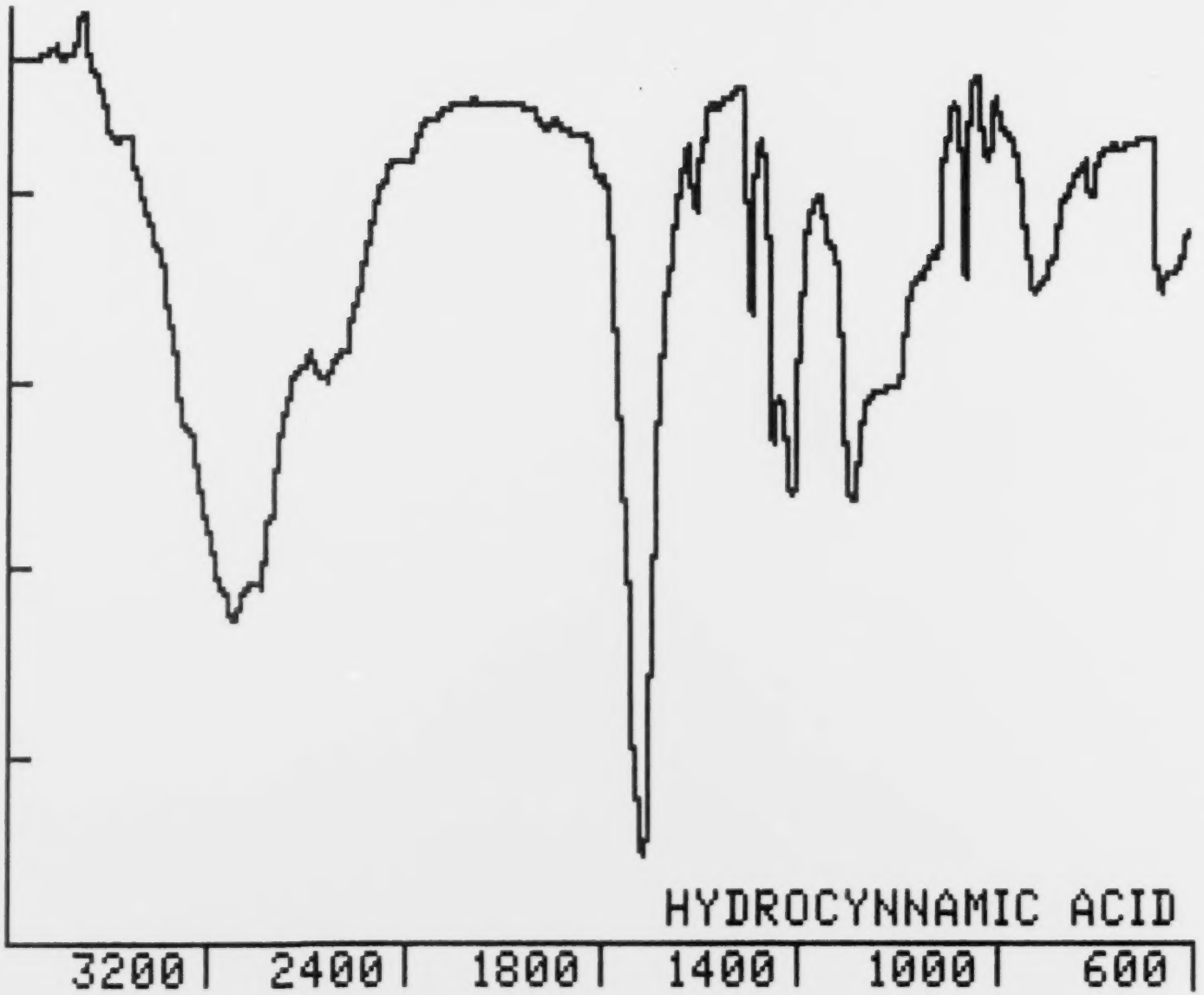
```
6080  GOSUB 15000: REM  Viewing Instructions
6090 :
6100  PRINT D$;"BLOAD ";FILE$;",A$4000"
6110 :
6120  GOSUB 9000: REM  Turn on Hires 2
6130 :
6140  RETURN
6150 :
7000  REM  ------------------------------------------------------------
7010  REM  !                    6.  Load data from text file           !
7020  REM  ------------------------------------------------------------
7030 :
7040  HOME
7050 :
7060  GOSUB 14000: REM  Get file name
7065 :
7070  GOSUB 15000: REM  Viewing Instructions
7080 :
7090  PRINT D$;"VERIFY ";FILE$
7100  PRINT D$;"OPEN ";FILE$
7110  PRINT D$;"READ ";FILE$
7120 :
7130  FOR I = 1 TO NUMBER
7140  INPUT Z(I)
7150  NEXT I
7160 :
7170  PRINT D$;"CLOSE ";FILE$
7180 :
7190  GOSUB 12000: REM  Plot data
7200 :
7210  GET KEY$
7220 :
7230  RETURN
7240 :
8000  REM  ------------------------------------------------------------
8010  REM  !                       7.  Print routine                   !
8020  REM  ------------------------------------------------------------
8030 :
8040  HOME
8050 :
8060  VTAB 12: HTAB 07: PRINT "Make sure printer is on line"
8070  VTAB 14: HTAB 07: PRINT "Press any key to print ";: GET KEY$
8080 :
8090  PRINT
8100  PRINT D$;"PR#1"
8110  PRINT I$;"G2DR"
8120  PRINT D$;"PR#0"
8130  PRINT
8140 :
8150  RETURN
8160 :
9000  REM  ------------------------------------------------------------
9010  REM  !                    8.  Turn on hires page 2               !
9020  REM  ------------------------------------------------------------
9030 :
9040  POKE 49232,0
9050  POKE 49234,0
9060  POKE 49237,0
9070  POKE 49239,0
9080 :
```

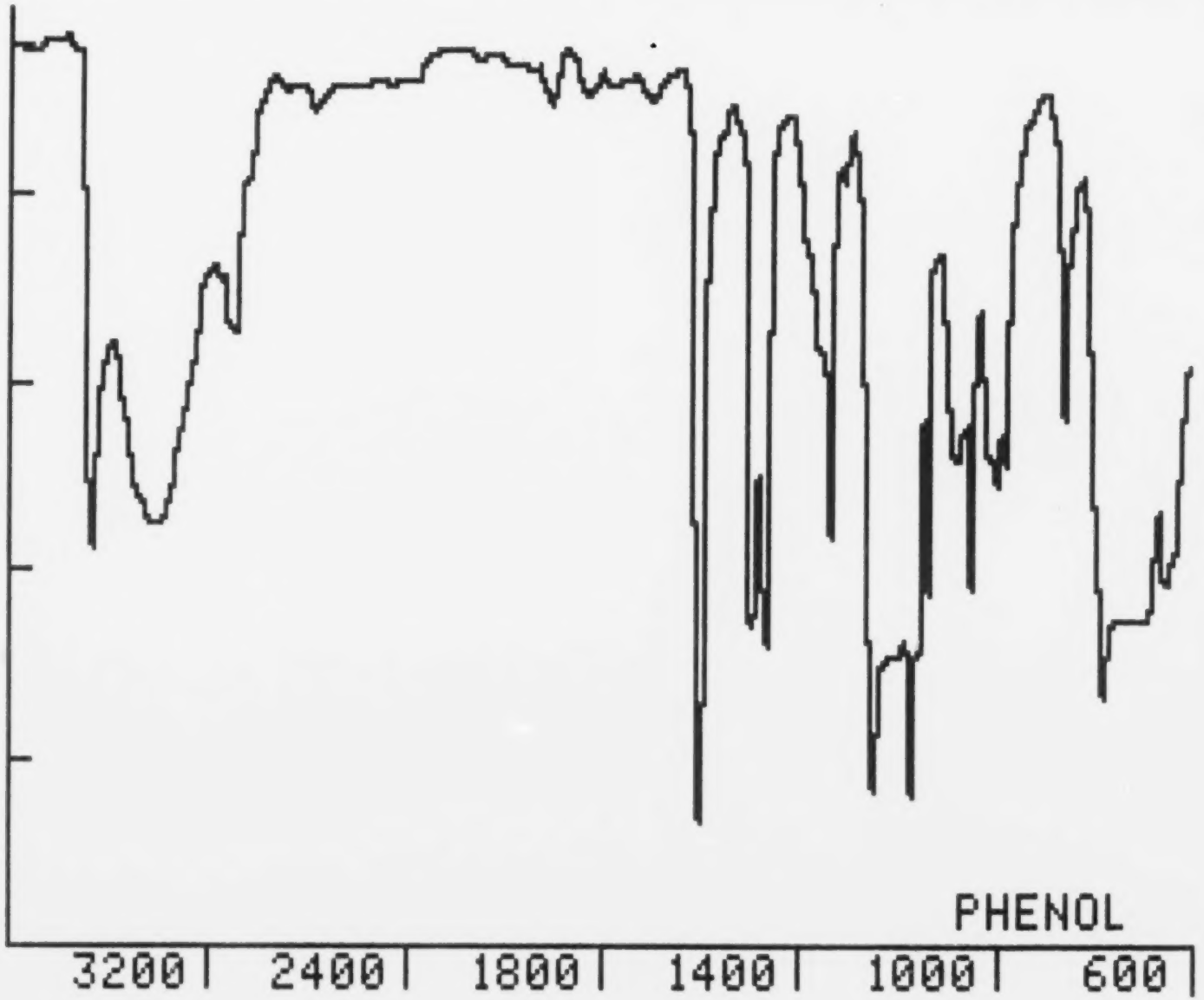
```
9090  GET KEY$
9100 :
9110  RETURN
9120 :
10000  REM  ------------------------------------------------------------
10010  REM  !                          Title Page                        !
10020  REM  ------------------------------------------------------------
10030 :
10040  HOME
10050 :
10060  VTAB 06: HTAB 11: PRINT "John Abbott College"
10070  VTAB 10: HTAB 08: PRINT "Infrared Spectrum Plotter"
10080  VTAB 13: HTAB 20: PRINT "by"
10090  VTAB 15: HTAB 17: PRINT "Jim Lyons"
10100  VTAB 17: HTAB 14: PRINT "Manoj Verghese"
10110 :
10120  FOR I = 1 TO 2000: NEXT I: REM  Pause
10130 :
10140  RETURN
10150 :
11000  REM  ------------------------------------------------------------
11010  REM  !                         Display Menu                       !
11020  REM  ------------------------------------------------------------
11030 :
11040  TEXT : HOME
11050 :
11060  PRINT "                  --  Menu --           ": PRINT
11070  PRINT "      Instructions ................. 1": PRINT
11080  PRINT "      Run scan ..................... 2": PRINT
11090  PRINT "      Save graph to binary file .... 3": PRINT
11100  PRINT "      Save data to text file ....... 4": PRINT
11110  PRINT "      Load graph from disk ......... 5": PRINT
11120  PRINT "      Load data from disk .......... 6": PRINT
11130  PRINT "      Print graph in memory ........ 7": PRINT
11140  PRINT "      View graph in memory ......... 8": PRINT
11150  PRINT "      Quit ......................... 9": PRINT
11160 :
11170  HTAB 1: VTAB 22
11180  PRINT "             Enter selection:  ";: GET AN$
11190 :
11200  IF AN$ < "1" OR AN$ > "9" THEN 11170
11210 :
11220  RETURN
11230 :
12000  REM  ------------------------------------------------------------
12010  REM  !                          Screen Plot                       !
12020  REM  ------------------------------------------------------------
12030 :
12040  HGR2 : HCOLOR= 3
12050 :
12060  PRINT
12070  PRINT  CHR$ (4);"BLOAD AXIS.IR,A$4000,D1"
12080  PRINT
12090 :
12100  FOR I = 1 TO NUMBER - 4 STEP 4
12120 X1 = (I - 1) / 4
12125 X2 = (I + 4 - 1) / 4
12130 Y1 = 182 - Z(I) * 181 / 4095
12135 Y2 = 182 - Z(I + 4) * 181 / 4095
12140  HPLOT X1,Y1 TO X2,Y2
```

```
12150  NEXT I
12160 :
12170  RETURN
12180 :
13000  REM  ------------------------------------------------------------
13010  REM  |                        Shows stats on data                 |
13020  REM  ------------------------------------------------------------
13030 :
13040  HOME
13050 :
13060  VTAB 08: HTAB 14: PRINT "Minimum : ";MIN
13070  VTAB 10: HTAB 14: PRINT "Maximum : ";MAX
13080  VTAB 12: HTAB 14: PRINT "Range   : ";MAX - MIN + 1
13090 :
13100 MIN = 4096
13110 MAX = 0
13120 :
13130  VTAB 17: HTAB 10
13140  PRINT "Press any key to continue ";: GET KEY$
13150 :
13160  RETURN
13170 :
14000  REM  ------------------------------------------------------------
14010  REM  |                          File name                         |
14020  REM  ------------------------------------------------------------
14030 :
14040  HOME
14050 :
14060  PRINT D$
14070  PRINT D$;"CATALOG,D2"
14080  PRINT
14090  PRINT
14100 :
14110  HTAB 1: VTAB 24
14120  INPUT "File name :  ";FILE$
14130  IF FILE$ = "" THEN 14110
14140 :
14150  RETURN
14160 :
15000  REM  ------------------------------------------------------------
15010  REM  |                       Viewing Instructions                 |
15020  REM  ------------------------------------------------------------
15030 :
15040  HOME
15050 :
15060  VTAB 06: HTAB 12: PRINT "Loading ";FILE$
15070  VTAB 08: HTAB 07: PRINT "When you have viewed the plot"
15080  VTAB 10: HTAB 09: PRINT "Press any key to continue"
15090 :
15100  RETURN
15110 :

]
```


BENZOIC ACID
3200
2400
1800
1400
1000
600


HYDROCYNNAMIC ACID
3200
2400
1800
1400
1000
600


PHENOL
3200
2400
1800
1400
1000
600

NMR.PLOT

Nmr.Plot

Introduction

"Nmr.plot" is a program which runs with the nuclear magnetic resonance spectrophotometer. This program is meant to replace the chart recorder and integrator that are found on the Nmr. "Nmr.plot" not only produces spectra but can save the graphs and data for later reference and manipulation. Futhermore, the "Graphics Writer" program (with modification) can be used to label the graphs once "Nmr.plot" has produced them.

Requirements

"Nmr.plot" will run on any Apple II series computer with a minimum of 64 k memory , an 80 column card and two disk drives. A twelve bit analog to digital converter in slot#7 is needed to convert the incoming voltage from the Nmr. An "Applied Engineering" converter was used for our project. For more information please refer to the "12 bit A/D Users manual". If hardcopies of pictures are desired a graphics supporting printer and an interface card in slot#1 is needed.

Two other files are needed on the disk to run "Nmr.plot". The "Axis.nmr" binary file provides a set of labeled axes for spectrum to be drawn on. Page one of the axis contains 10 to 5, page two contains 5 to 0 of the x axis. The "Dos Mover" binary program (Cornelis Bonger, Micro-Sparc Inc.) moves dos from its usual location ($9600 - $C000) to the top of memory ($D000 - $FFFF). This makes available an extra and necessary 10 k for the program and its variables. Booting the disk executes "Boot.system" which runs "Dos Mover" that runs "Hello". This leaves you to run "Nmr.plot". If the disk is not booted, you can of course, brun "Dos Mover" from the keyboard.

Main Program: 1380 - 1540

At startup the title and programmer's name are displayed on the text screen. After a brief pause, the variables are initialized. The menu is then displayed. The program waits for the user to select one of the nine options. After having completed the execution of an option the program returns to the menu.

Initialization of Variables: 1550 - 1850

The program variables are set to their initial values. Please refer to the "Dictionary of Variables" found at lines 1100 to 1370 for the definitions.

1. Instructions: 2000 - 2070

The instruction module for this program has not been written.

2. Data Collection: 3000 - 3450

A message is displayed instructing the user to start the Nmr scan. The Nmr scan takes a little over two minutes. The recorder (pen) must be pushed to the extreme left where the it is at the 11 mark. The 'return' and 'scan' buttons should be pressed simultaneously for best results. Sampling will begin at 10 and end at 0. A total of 558 points of data are collected. During sampling the most and least significant bytes are stored. After sampling, they are converted to integer values. The minimum, maximum and range of data are shown. The spectrum is drawn on pages one and two for which the integration is also plotted. By pressing the spacebar the display is toggled between pages one and two. Pressing any other key will bring up the menu.

3. Save Hires pages to binary file: 4000 - 4090

Drive 2 is catalogued and the user is asked for a filename. Both hi-res pages are saved to disk as one file (A$2000,L$3FFF). The user is responsible for ensuring that there is sufficient disk space (a graph uses 66 blocks) and a locked file of the same name does not exist.

4. Save data array to text file: 5000 - 5180

Drive 2 is catalogued and the user is asked for a filename. The sequential textfile is opened and the data is recorded. The user is responsible for ensuring that there is sufficient disk space and a locked file of the same name does not exist.

5. Load graph from disk: 6000 - 6150

Drive 2 is catalogued and the user is asked for a filename. The user is told of the viewing instructions. The binary file is loaded onto pages one and two. The display switches to page one. Pressing the spacebar toggles between the two pages, any other key returns to the menu.

6. Load data from text file: 7000 - 7250

Drive 2 is catalogued and the user is asked for a filename. The user is told of the viewing instructions. The text file is first verified then opened and read. The graph is plotted. Pressing the spacebar toggles between the two pages, any other key returns to the menu.

7. Print Routine: 8000 - 8170

The user is told to set up the printer and press any key when ready. Hi-res pages one and two are rotated and dumped side by side. The interface card is assumed to be a Grappler+ installed in slot #1.

8. Turn on hires page 1: 9000 - 9120

Hi-res page one is displayed. The user presses the spacebar to toggle the display between pages one and two. Pressing any other key causes the program to return to the menu.

Title Page: 10 000 - 10 290

The program title and programmers' names are displayed. After a brief pause, the program continues.

Display Menu: 11 000 - 11 250

The menu consisting of nine options is shown. The user selects one of the nine. If the number chosen is invalid the user is prompted again.

Screen Plot: 12 000 - 12 380

The display is switched to hi-res page one. The color is set to white. The predrawn axes "Axis.nmr" is loaded onto pages one and two. The 558 points collected are displayed on two pages (279 x 2). Page one, from 10 to 5 is drawn first, then page two, from 5 to 0. An integration for the spectrum is also drawn.

Show stats on data: 13 000 - 13 170

The minimum, maximum values and the range of data is shown. These values are then reset. The user is asked to press any key to continue.

File name: 14 000 - 14 160

Drive 2 is catalogued and the user is asked for a file name. If no name is entered the user is prompted again. Invalid file names are not checked for (eg. one starting with a number). The user is responsible for specifying the correct file when requested.

Viewing Instructions: 15 000 - 15 130

The user is told that the file specified is being loaded. Once the graph is drawn, pressing the spacebar will toggle the display between pages one and two. Pressing any other key will cause the progam to return to the menu.

Hi-Res Graphics Page Toggle: 16 000 - 16 110

If the user presses the spacebar, the display is switched from the current page being shown to the other page (between one and two). Pressing any other key causes the program to return to the menu.

Integration: 17 000 - 17 370

Integration is accomplished by simply summing the y values (data). Care has been taken to see that the plot stays within the limits of the hi-res page.

```
]LIST

10  IF  PEEK (103) +  PEEK (104) * 256 = 24577 THEN 70
20  POKE 24576,0
30  POKE 103,1
40  POKE 104,96
50  PRINT  CHR$ (4);"RUN NMR.PLOT"
60  PRINT  CHR$ (4);
70  PRINT  CHR$ (4);"PR#3"
80  PRINT
90 :
1000  REM  ------------------------------------------------------------
1010  REM  !       Nuclear Magnetic Resonance Spectrum Plotter          !
1020  REM  ------------------------------------------------------------
1030 :
1040  REM  Date       :    June 17, 1988
1050  REM  Program    :    NMRplot   ( modified from GCplot.3 )
1060  REM  Authors    :    Jim Lyons      - original version (Gc)
1070  REM                  Manoj Verghese - modification for NMR
1080  REM                                 - listing format
1090 :
1100  REM  ------------------------------------------------------------
1110  REM  !                  Dictionary of Variables                  !
1120  REM  ------------------------------------------------------------
1130 :
1140  REM  Channel     :  channel of card for data sampling
1150  REM  Gain        :  gain setting
1160  REM  Cslot       :  slot number of card
1170  REM  Addr        :  address of card
1180 :
1190  REM  Number      :  number of samples taken
1200  REM  z(i)        :  array to store samples
1210  REM  za          :  z(i) is given average value of za
1220 :
1230  REM  Max         :  maximum value from samples
1240  REM  Min         :  minimum value from samples
1250 :
1260  REM  D$          :  control - d
1270  REM  I$          :  control - i
1280 :
1290  REM  An$         :  user selected menu option
1300  REM  Key$        :  user hits any key to continue
1310  REM  File$       :  name of current graph or text file
1320 :
1330  REM  i, j        :  loop counters - indices for z
1340 :
1350  REM  x, y1       :  start point of line being plotted
1360  REM  x+1, y2     :  end point of line being plotted
1370 :
1380  REM  ------------------------------------------------------------
1390  REM  !                       Main Execution                      !
1400  REM  ------------------------------------------------------------
1410 :
1420  GOSUB 10000: REM  Title Page
1430 :
1440  GOSUB 1550: REM  Initialize variables
1450 :
1460  GOSUB 11000: REM  Menu
1470 :
```

```
1480  ON  VAL (AN$) GOSUB 2000,3000,4000,5000,6000,7000,8000,9000
1490 :
1500  IF AN$ <  > "9" THEN 1460
1510 :
1520  HOME
1530  END
1540 :
1550  REM  ------------------------------------------------------------
1560  REM  !                    Initialization of variables             !
1570  REM  ------------------------------------------------------------
1580 :
1590 :
1600 CHANNEL = 0
1610 GAIN = 3
1620 CSLOT = 7
1630 ADDR = 49280 + CSLOT * 16
1640 :
1650 NUMBER = 558
1660  DIM Z(NUMBER * 2)
1670 ZA = 0
1680 :
1690 OVER = 4090
1700 MAX = 0
1710 MIN = 4096
1720 :
1730 D$ =  CHR$ (4)
1740 I$ =  CHR$ (9)
1750 :
1760 AN$ = ""
1770 KEY$ = ""
1780 FILE$ = ""
1790 :
1800 I = 0
1810 J = 0
1820 :
1830 X1 = 0:Y1 = 0
1840 X2 = 0:Y2 = 0
1850 :
2000  REM  ------------------------------------------------------------
2010  REM  !                  1.  Instructions Routine                  !
2020  REM  ------------------------------------------------------------
2030 :
2040  REM  Sorry, no instruction file is ready
2050 :
2060  RETURN
2070 :
3000  REM  ------------------------------------------------------------
3010  REM  !                    2.  Data Collection                     !
3020  REM  ------------------------------------------------------------
3030 :
3040  REM  ..................... sampling message .....................
3050 :
3060  HOME
3070 :
3080  VTAB 07: HTAB 08: PRINT "Prepare spectrophotometer"
3090  VTAB 09: HTAB 08: INPUT "When ready press return ";KEY$
3100  VTAB 13: HTAB 12: PRINT "Sampling Data ..."; CHR$ (7)
3110 :
3120  FOR I = 1 TO 3500: NEXT I: REM  synchronize start
3130 :
```

```
3140  REM  ..................... read a/d card .....................
3150 :
3160  POKE ADDR + 3,GAIN * 16 + CHANNEL: REM  initializes a/d card
3170 :
3180  FOR I = 1 TO NUM
3190 :
3200  FOR J = 1 TO 110: NEXT J: REM  2 min 10 sec scan
3210 :
3220  POKE ADDR + 2,0
3240 Z(I) =  PEEK (ADDR)
3250 Z(I + NUM) =  PEEK (ADDR + 1)
3260 :
3270  NEXT I
3280 :
3290  REM  ................. analysis and display of data ..............
3300 :
3310  VTAB 13: HTAB 12: PRINT "Processing data ..."; CHR$ (7)
3320 :
3330  FOR I = 1 TO NUM
3332 Z(I) = Z(I) + Z(I + NUM) * 256
3340  IF Z(I) > MAX THEN MAX = Z(I)
3350  IF MIN > Z(I) THEN MIN = Z(I)
3360  NEXT I
3370 :
3380  GOSUB 13000: REM  show statistics on data
3390 :
3400  GOSUB 12000: REM  screen plot
3410 :
3420  GOSUB 16000: REM  Toggle hi-res pages
3430 :
3440  RETURN
3450 :
4000  REM  ------------------------------------------------------------
4010  REM  !            3.  Save Hires pages to binary file           !
4020  REM  ------------------------------------------------------------
4030 :
4040  GOSUB 14000: REM  Get file name
4050 :
4060  PRINT D$;"BSAVE ";FILE$;",A$2000,L$3FFF"
4070 :
4080  RETURN
4090 :
5000  REM  ------------------------------------------------------------
5010  REM  !             4.  Save data array to text file             !
5020  REM  ------------------------------------------------------------
5030 :
5040  HOME
5050 :
5060  GOSUB 14000: REM  Get file name
5070 :
5080  PRINT D$;"OPEN ";FILE$
5090  PRINT D$;"WRITE ";FILE$
5100 :
5110  FOR I = 1 TO NUMBER
5120  PRINT Z(I)
5130  NEXT I
5140 :
5150  PRINT D$;"CLOSE ";FILE$
5160 :
5170  RETURN
```

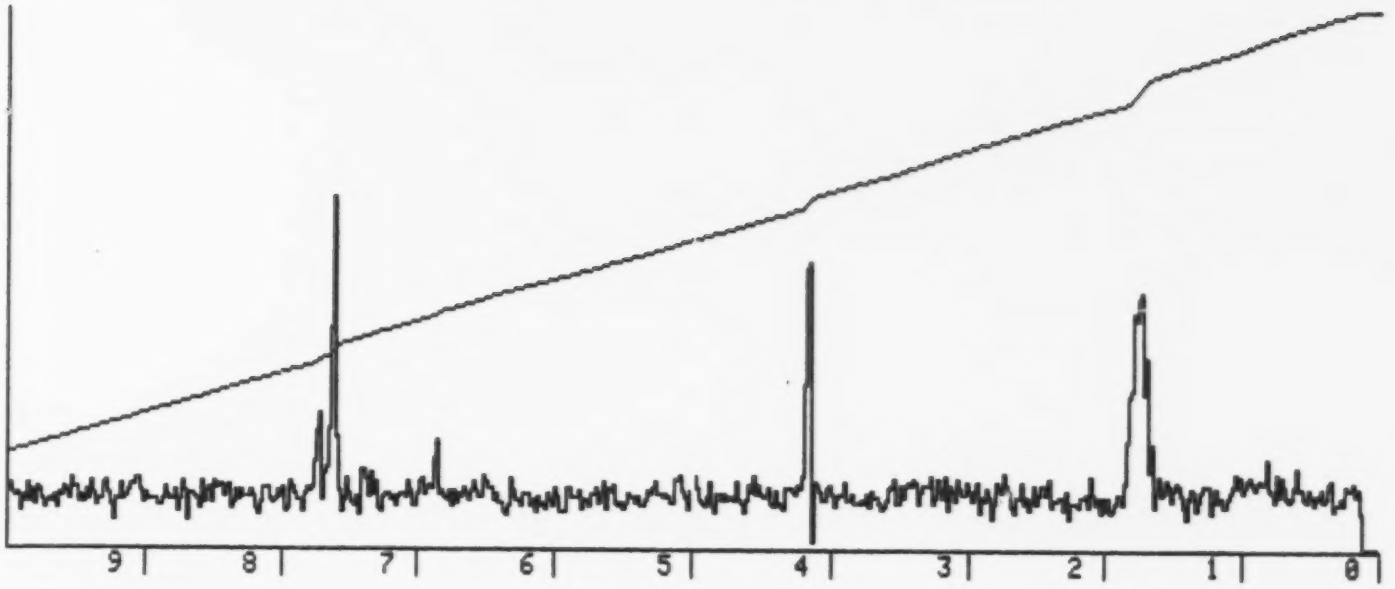
```
5180 :
6000  REM  ------------------------------------------------------------
6010  REM  !                  5.  Load graph from disk                !
6020  REM  ------------------------------------------------------------
6030 :
6040  HOME
6050 :
6060  GOSUB 14000: REM  Get file name
6070 :
6080  GOSUB 15000: REM  Viewing Instructions
6090 :
6100  PRINT D$;"BLOAD ";FILE$;",A$2000"
6110 :
6120  GOSUB 9000: REM  Turn on Hires 1
6130 :
6140  RETURN
6150 :
7000  REM  ------------------------------------------------------------
7010  REM  !                  6.  Load data from text file            !
7020  REM  ------------------------------------------------------------
7030 :
7040  HOME
7050 :
7060  GOSUB 14000: REM  Get file name
7070 :
7080  GOSUB 15000: REM  Viewing Instructions
7090 :
7100  PRINT D$;"VERIFY ";FILE$
7110  PRINT D$;"OPEN ";FILE$
7120  PRINT D$;"READ ";FILE$
7130 :
7140  FOR I = 1 TO NUMBER
7150  INPUT Z(I)
7160  NEXT I
7170 :
7180  PRINT D$;"CLOSE ";FILE$
7190 :
7200  GOSUB 12000: REM  Plot data
7210 :
7220  GOSUB 16000: REM  Toggle hi-res pages
7230 :
7240  RETURN
7250 :
8000  REM  ------------------------------------------------------------
8010  REM  !                    7.  Print routine                     !
8020  REM  ------------------------------------------------------------
8030 :
8040  HOME
8050 :
8060  VTAB 12: HTAB 07: PRINT "Make sure printer is on line"
8070  VTAB 14: HTAB 07: PRINT "Press any key to print ";: GET KEY$
8080 :
8090  PRINT
8100  PRINT  CHR$ (4);"PR#1"
8110  PRINT  CHR$ (9);"GR"
8120  PRINT  CHR$ (9);"G2R"
8130  PRINT  CHR$ (4);"PR#3"
8140  PRINT
8150 :
8160  RETURN
```

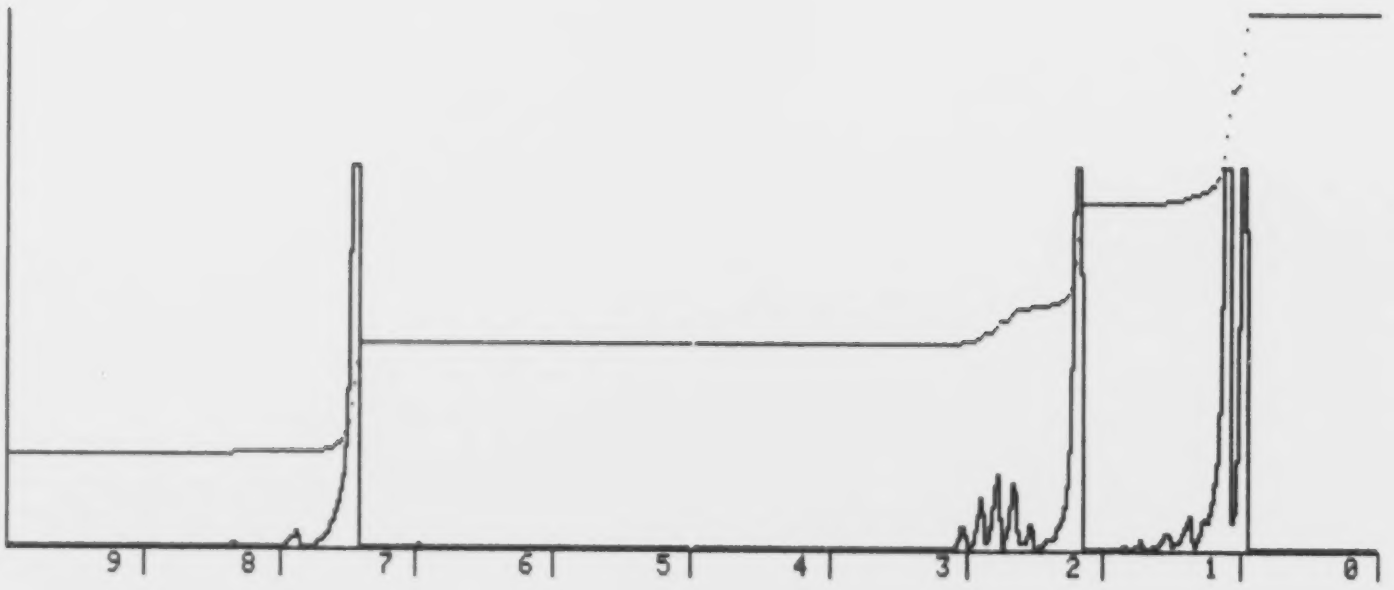
```
8170 :
9000  REM  ------------------------------------------------------------
9010  REM  !                  8.  Turn on hires page 1                  !
9020  REM  ------------------------------------------------------------
9030 :
9040  POKE 49232,0
9050  POKE 49234,0
9060  POKE 49236,0
9070  POKE 49239,0
9080 :
9090  GOSUB 16000: REM  Toggle Hi-res pages
9100 :
9110  RETURN
9120 :
10000  REM  ------------------------------------------------------------
10010  REM  !                          Title Page                        !
10020  REM  ------------------------------------------------------------
10030 :
10040  PRINT  CHR$ (4)
10050  PRINT  CHR$ (4);"PR#3"
10060  PRINT
10070 :
10080  FOR I = 8 TO 74
10090  VTAB 01: POKE 1403,I: PRINT "_"
10100  VTAB 21: POKE 1403,I: PRINT "_"
10110  NEXT I
10120 :
10130  FOR I = 2 TO 21
10140  VTAB I: POKE 1403,06: PRINT "!"
10150  VTAB I: POKE 1403,75: PRINT "!"
10160  NEXT I
10170 :
10180  VTAB 04: POKE 1403,25: PRINT "     John Abbott College Cegep     "
10190  VTAB 07: POKE 1403,25: PRINT "Manoj Verghese and Beagle Brothers"
10200  VTAB 09: POKE 1403,25: PRINT "            present for            "
10210  VTAB 11: POKE 1403,25: PRINT "      The Chemistry Department     "
10220  VTAB 14: POKE 1403,25: PRINT "     Nuclear Magnetic Resonance    "
10230  VTAB 16: POKE 1403,25: PRINT "          Spectrum Display         "
10240  VTAB 18: POKE 1403,25: PRINT "    In High Resolution Graphics    "
10250 :
10260  FOR I = 1 TO 3000: NEXT I
10270 :
10280  RETURN
10290 :
11000  REM  ------------------------------------------------------------
11010  REM  !                        Display Menu                        !
11020  REM  ------------------------------------------------------------
11030 :
11040  TEXT : PRINT  CHR$ (4): PRINT  CHR$ (4);"PR#3": PRINT
11050 :
11060  VTAB 01: POKE 1403,25: PRINT "     M          "
11070  VTAB 04: POKE 1403,25: PRINT "Instructions ..................... 1"
11080  VTAB 06: POKE 1403,25: PRINT "Run scan ......................... 2"
11090  VTAB 08: POKE 1403,25: PRINT "Save spectrum .................... 3"
11100  VTAB 10: POKE 1403,25: PRINT "Save data ........................ 4"
11110  VTAB 12: POKE 1403,25: PRINT "Load spectrum .................... 5"
11120  VTAB 14: POKE 1403,25: PRINT "Read data file ................... 6"
11130  VTAB 16: POKE 1403,25: PRINT "Dump spectrum .................... 7"
11140  VTAB 18: POKE 1403,25: PRINT "View current spectrum ........... 8"
11150  VTAB 20: POKE 1403,25: PRINT "Exit program ..................... 9"
```

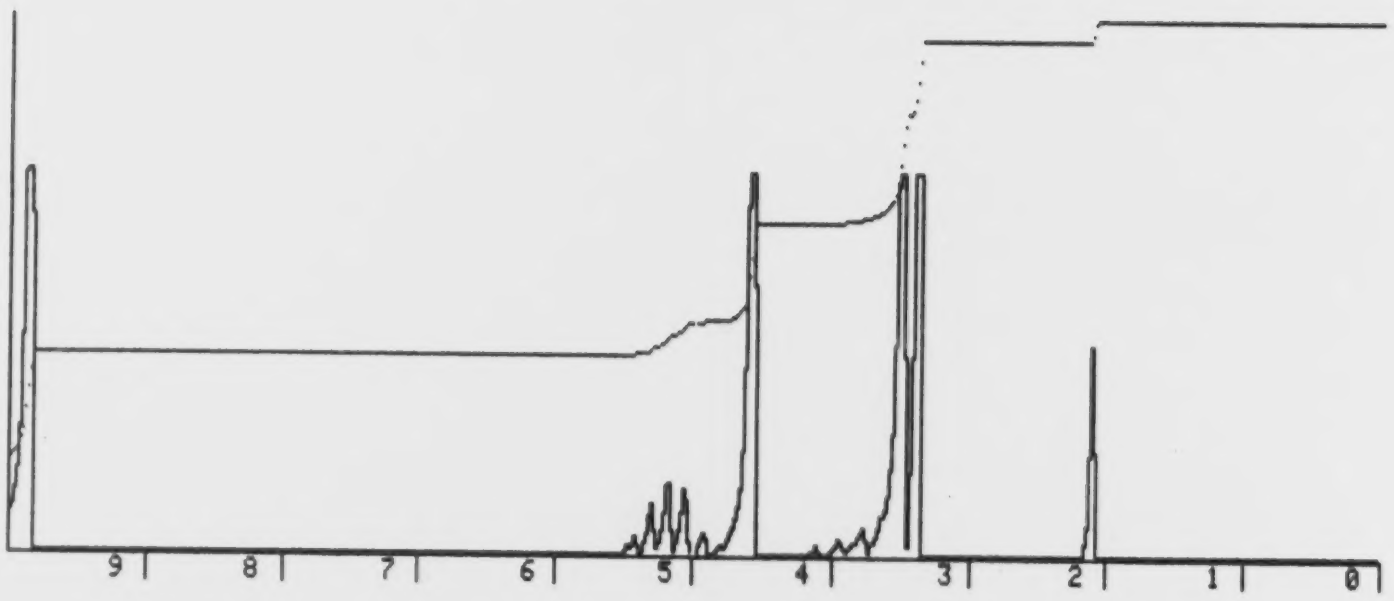
```
11160 :
11170  VTAB 23: POKE 1403,25: PRINT "        Enter selection : ";
11180  GET AN$
11190 :
11200  IF AN$ < "1" OR AN$ > "9" THEN 11170
11210 :
11220  PRINT  CHR$ (27); CHR$ (17): REM  40 columns
11230 :
11240  RETURN
11250 :
12000  REM  ------------------------------------------------------------
12010  REM  !                        Screen Plot                        !
12020  REM  ------------------------------------------------------------
12030 :
12040  HGR2 : HCOLOR= 3
12050 :
12060  PRINT
12070  PRINT  CHR$ (4);"BLOAD AXIS.NMR,A$2000,D1"
12080  PRINT
12090 :
12100  POKE 49236,0: REM  Turn on page 1
12110  POKE 230,32: REM  Enable drawing on page 1
12120 X = 0
12130 :
12140  FOR I = 1 TO NUM / 2 - 1
12150  GOSUB 12300
12160  NEXT I
12170 :
12180  POKE 49237,0: REM  Turn on page 2
12190  POKE 230,64: REM  Enable drawing on page 2
12200 X = 0
12210 :
12220  FOR I = NUM / 2 + 1 TO NUM - 1
12230  GOSUB 12300
12240  NEXT I
12250 :
12260  GOSUB 17000: REM  Integration
12270 :
12280  RETURN
12290 :
12300  REM  ..................... plotting points .....................
12310 :
12320 X = X + 1
12330 Y1 = 181 - Z(I) * 180 / 4095
12340 Y2 = 181 - Z(I + 1) * 180 / 4095
12350  HPLOT X,Y1 TO X,Y2
12360 :
12370  RETURN
12380 :
13000  REM  ------------------------------------------------------------
13010  REM  !                    Shows stats on data                    !
13020  REM  ------------------------------------------------------------
13030 :
13040  HOME
13050 :
13060  VTAB 08: HTAB 14: PRINT "Minimum : ";MIN
13070  VTAB 10: HTAB 14: PRINT "Maximum : ";MAX
13080  VTAB 12: HTAB 14: PRINT "Range   : ";MAX - MIN + 1
13090 :
13100 MAX = 0
```

```
13110 MIN = 4096
13120 :
13130  VTAB 17: HTAB 10
13140  NORMAL : PRINT "Press any key to continue ";: GET KEY$
13150 :
13160  RETURN
13170 :
14000  REM  ----------------------------------------------------------
14010  REM  !                          File name                       !
14020  REM  ----------------------------------------------------------
14030 :
14040  HOME
14050 :
14060  PRINT D$
14070  PRINT D$;"CATALOG,D2"
14080  PRINT
14090  PRINT
14100 :
14110  HTAB 1: VTAB 24
14120  INPUT "File name : ";FILE$
14130  IF FILE$ = "" THEN 14110
14140 :
14150  RETURN
14160 :
15000  REM  ----------------------------------------------------------
15010  REM  !                     Viewing Instruction                  !
15020  REM  ----------------------------------------------------------
15030 :
15040  HOME
15050 :
15060  VTAB 04: HTAB 12: PRINT "Loading ";FILE$
15070  VTAB 08: HTAB 08: PRINT "The spacebar toggles between"
15080  VTAB 10: HTAB 08: PRINT "the 2 hi-res graphics pages."
15090  VTAB 13: HTAB 08: PRINT "Any other key returns to the"
15100  VTAB 15: HTAB 08: PRINT "main menu.                  "
15110 :
15120  RETURN
15130 :
16000  REM  ----------------------------------------------------------
16010  REM  !                Hi-Res Graphics Page Toggle               !
16020  REM  ----------------------------------------------------------
16030 :
16040  GET KEY$
16050 :
16060  IF KEY$ <  > " " THEN  RETURN
16070 :
16080  POKE 49236 + PAGE,0
16090 PAGE = 1 - PAGE
16100  GOTO 16040
16110 :
17000  REM  ----------------------------------------------------------
17010  REM  !                          Integration                     !
17020  REM  ----------------------------------------------------------
17030 :
17040  FOR I = 1 TO NUM
17050 Y = Y + Z(I)
17060  NEXT I
17070 :
17080 FUDGE = Y / 150
17090 :
```

```
17100  POKE 49236,0: REM  Turn on page 1
17110  POKE 230,32: REM  Enable drawing on page 1
17120 X = 0:Y = 0
17130 :
17140  FOR I = 1 TO NUM / 2
17150  GOSUB 17300: REM  Get co-ordinates
17160  NEXT I
17170 :
17180  POKE 49237,0: REM  Turn on page 2
17190  POKE 230,64: REM  Enable drawing on page 2
17200 X = 0
17210 :
17220  FOR I = NUM / 2 + 1 TO NUM
17230  GOSUB 17300: REM  Get co-ordinates
17240  NEXT I
17250 :
17260 X = 0:Y = 0
17270 :
17280  RETURN
17290 :
17300  REM  ..... co-ordinates .....
17310 :
17320 X = X + 1
17330 Y = Y + Z(I)
17340 :
17350  HPLOT X,150 - Y / FUDGE
17360 :
17370  RETURN
```


9
8
7
6
5
4
3
2
1
0


9
8
7
6
5
4
3
2
1
0


9
8
7
6
5
4
3
2
1
0

NMR.PLOT.DHGR

Nmr.Plot.Dhgr

Introduction

"Nmr.plot.dhgr" is a program which runs with the Nuclear Magnetic Resonance Spectrophotometer. This program is meant to replace the chart recorder and integrator that are found on the Nmr. "Nmr.plot.dhgr" not only produces spectra but can save the graphs and data for later reference and manipulation. This program is essentially the same as "Nmr.plot" except that the horizontal resolution is doubled and as a result the graph is displayed on one page rather than two as was the case with the earlier program.

Requirements

"Nmr.plot.dhgr" will run on an Apple IIc or a IIe with 128 k and two disk drives. A twelve bit analog to digital converter in slot#7 is needed to convert the incoming voltage from the Nmr. An "Applied Engineering" converter was used for our project. For more information please refer to the "12 bit A/D Users manual". If hardcopies of pictures are desired a graphics supporting printer and an interface card in slot#1 is needed.

Four other files are needed on the disk to run "Nmr.plot.dhgr". The "Axis.nmr.dhgr", "Axis.nmr.dhgr.aux" binary files provides a set of labeled axes for spectrum to be drawn on.

The "Dos Mover" binary program (Cornelis Bonger, Micro-Sparc Inc.) moves Dos from its usual location ($9600 - $C000) to the top of memory ($D000 - $FFFF). This makes available an extra and necessary 10 k for the program and its variables. Booting the disk executes "Boot.system" which runs "Dos Mover" that runs "Hello". This leaves you to brun "Dhgr" and run "Nmr.plot.dhgr". If the disk is not booted, you can of course, do the above from the keyboard.

The "Dhgr" binary program (Mark Simonsen) which enables double hi-res must be brun before running "Nmr.plot.dhgr". For more information please refer to the "Beagle Graphics" manual.

Main Program: 1410 - 1570

At startup the title and programmer's name are displayed on the text screen. After a brief pause, the variables are initialized. The menu is then displayed. The program waits for the user to select one of the nine options. After having completed the execution of an option the program returns to the menu.

Initialization of Variables: 1580 - 1890

The program variables are set to their initial values. Please refer to the "Dictionary of Variables" found at lines 1100 to 1390 for the definitions.

1. Instructions: 2000 - 2070

The instruction module for this program has not been written.

2. Data Collection: 3000 - 3450

A message is displayed instructing the user to start the Nmr scan. The Nmr scan takes a little over two minutes. The recorder (pen) must be pushed to the extreme left where the it is at the 11 mark. The 'return' and 'scan' buttons should simultaneously for the best results. Sampling will begin at 10 and end at 0. A total of 558 points of data are collected. During sampling the most and least significant bytes are stored. After sampling they are converted to integer values. The minimum, maximum and range of data are shown. The spectrum is drawn on the double hi-res page for which the integration is also plotted. Pressing any key will bring up the menu.

3. Save Double Hires to binary file: 4000 - 4090

Drive 2 is catalogued and the user is asked for a filename. The double hi-res page is saved to disk as two files (pic, pic.aux). The user is responsible for ensuring that sufficient disk space (a graph uses 68 blocks) and a locked file of the same name does not exist.

4. Save data array to text file: 5000 - 5180

Drive 2 is catalogued and the user is asked for a filename. The sequential textfile is opened and the data is recorded. The user is responsible for ensuring that there is sufficient disk space and a locked file of the same name does not exist.

5. Load graph from disk: 6000 - 6170

Drive 2 is catalogued and the user is asked for a filename. The user is told of the viewing instructions. The binary files are loaded onto the double hi-res page. After viewing the graph, pressing any key returns to the menu.

6. Load data from text file: 7000 - 7230

Drive 2 is catalogued and the user is asked for a filename. The user is told of the viewing instructions. The text file is first verified then opened and read. The graph is plotted. Pressing any key returns to the menu.

7. Print Routine: 8000 - 8150

The user is asked to turn on the printer and press any key when ready. The interface card is assumed to support double hi-res graphics and be located slot#1.

8. Turn on Double hires: 9000 - 9090

The double hi-res page is displayed. Pressing any key causes the program to return to the menu.

Title Page: 10 000 - 10 290

The program title and programmers' names are displayed. After a brief pause, the program continues.

Display Menu: 11 000 - 11 230

The menu consisting of nine options is shown. The user selects one of the nine. If the number chosen is invalid the user is prompted again.

Screen Plot: 12 000 - 12 260

The display is switched to double hi-res The color is set to white. The predrawn axes "Axis.nmr.dhgr, Axis.nmr.dhgr.aux" is loaded from drive one. The 558 points collected are displayed on one page. An integration for the spectrum is also drawn.

Show stats on data: 13 000 - 13 170

The minimum, maximum values and the range of data is shown. These values are then reset. The user is asked to press any key to continue.

File name: 14 000 - 14 140

Drive 2 is catalogued and the user is asked for a file name. If no name is entered the user is prompted again. Invalid file names are not checked for (eg. one starting with a number). The user is responsible for specifying the correct file when requested.

Viewing Instructions: 15 000 - 15 120

The user is told that the file specified is being loaded. Once the graph is drawn, pressing any key will cause the progam to return to the menu.

Hi-Res Graphics Page Toggle: 16 000 - 16 120

If two pages were displayed instead of one, then a switch between pages would be needed. This could be located here. The routinue is left over from "Nmr.plot" which used two pages.

Integration: 17 000 - 17 230

Integration is accomplished by simply summing the y values. Care has been taken to see that the plot stays within the limits of the double hi-res page.

```
]LIST

10  IF  PEEK (103) +  PEEK (104) * 256 = 16385 THEN 60
20  POKE 16384,0
30  POKE 103,1
40  POKE 104,64
50  PRINT  CHR$ (4);"RUN NMR.PLOT.DHGR"
60 :
1000  REM  ------------------------------------------------------------
1010  REM  !         Nuclear Magnetic Resonance Spectrum Plotter       !
1020  REM  ------------------------------------------------------------
1030 :
1040  REM  Date       :    July 18, 1988
1050  REM  Program    :    NMR.PLOT.DHGR ( modified from Gcplot.3 )
1060  REM  Authors    :    Jim Lyons       - original version (Gc)
1070  REM                  Manoj Verghese - modification for NMR
1080  REM                                 - listing format
1090 :
1100  REM  ------------------------------------------------------------
1110  REM  !                    Dictionary of Variables                !
1120  REM  ------------------------------------------------------------
1130 :
1140  REM  Channel   :  channel of card for data sampling
1150  REM  Gain      :  gain setting for card
1160  REM  Cslot     :  slot number of card
1170  REM  Addr      :  address of card
1180 :
1190  REM  Number    :  number of samples taken
1200  REM  z(i)      :  array to store samples
1210  REM  Za        :  z(i) given average value of za
1220 :
1230  REM  Over      :  threshold value of sample readings
1240  REM  Max       :  maximum value from samples
1250  REM  Min       :  minimum value from samples
1260 :
1270  REM  D$        :  control - d
1280  REM  I$        :  control - i
1290 :
1300  REM  An$       :  user selected menu option
1310  REM  Key$      :  user hits any key to continue
1320  REM  File$     :  name of graphics or data file
1340 :
1350  REM  i,j       :  loop counters - indices of z
1360 :
1370  REM  x, y1     :  start point of line being plotted
1380  REM  x+1, y2   :  end point of line being plotted
1390 :
1400 :
1410  REM  ------------------------------------------------------------
1420  REM  !                       Main Execution                      !
1430  REM  ------------------------------------------------------------
1440 :
1450  GOSUB 10000: REM  Title Page
1460 :
1470  GOSUB 1580: REM  Initialize variables
1480 :
1490  GOSUB 11000: REM  Menu
1500 :
1510  ON  VAL (AN$) GOSUB 2000,3000,4000,5000,6000,7000,8000,9000
```

```
1520 :
1530  IF AN$ <  > "9" THEN 1490
1540 :
1550  HOME
1560  END
1570 :
1580  REM  ------------------------------------------------------------
1590  REM  !                    Initialization of variables             !
1600  REM  ------------------------------------------------------------
1610 :
1620 CHANNEL = 0
1630 GAIN = 3
1640 CSLOT = 7
1650 ADDR = 49280 + CSLOT * 16
1660 :
1670 NUMBER = 558
1680  DIM Z(NUMBER * 2)
1690 ZA = 0
1700 :
1710 OVER = 4090
1720 MAX = 0
1730 MIN = 4096
1740 :
1750 D$ =  CHR$ (4)
1760 I$ =  CHR$ (9)
1770 :
1780 AN$ = ""
1790 KEY$ = ""
1800 FILE$ = ""
1810 :
1820 I = 0
1830 J = 0
1840 :
1850 X1 = 0:Y1 = 0
1860 X2 = 0:Y2 = 0
1870 :
1880  RETURN
1890 :
2000  REM  ------------------------------------------------------------
2010  REM  !                  1.  Instructions Routine                  !
2020  REM  ------------------------------------------------------------
2030 :
2040  REM  Sorry, the instruction mode has not been written
2050 :
2060  RETURN
2070 :
3000  REM  ------------------------------------------------------------
3010  REM  !                     2.  Data Collection                    !
3020  REM  ------------------------------------------------------------
3030 :
3040  REM  ..................... sampling message .....................
3050 :
3060  HOME
3070 :
3080  VTAB 07: POKE 1403,24: PRINT "Prepare spectrophotometer"
3090  VTAB 09: POKE 1403,25: PRINT "When ready press return ";: GET KEY$
3100  VTAB 13: POKE 1403,26: PRINT "Sampling Data ..."; CHR$ (7)
3110 :
3120  FOR I = 1 TO 3500: NEXT I: REM  synchronize with Nmr
3130 :
```

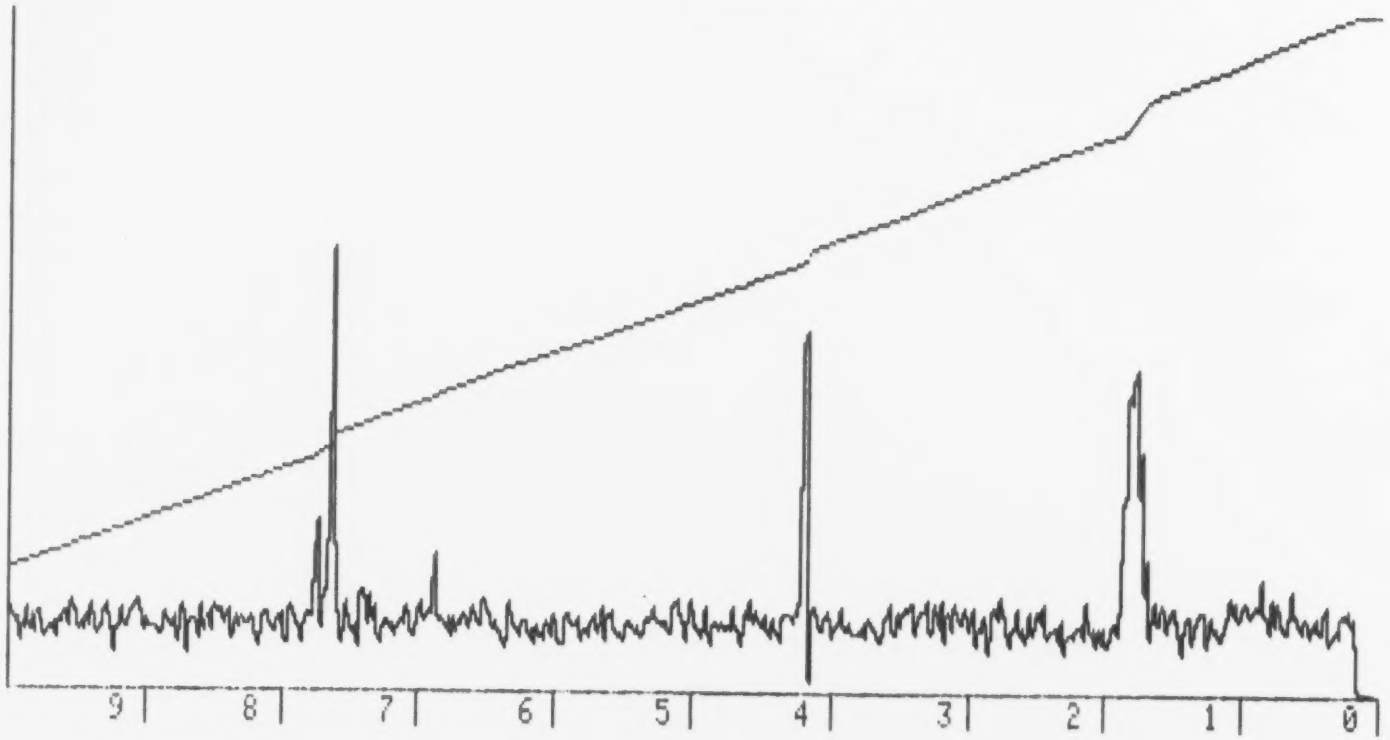
```
3140  REM  ..................... read a/d card .....................
3150 :
3160  POKE ADDR + 3,GAIN * 16 + CHANNEL: REM  initializes a/d card
3170 :
3180  FOR I = 1 TO NUM
3190 :
3200  FOR J = 1 TO 10: NEXT J: REM  synchronize with Nmr
3210 :
3220  POKE ADDR + 2,0
3240 Z(I) =  PEEK (ADDR)
3250 Z(I + NUM) =  PEEK (ADDR + 1)
3260 :
3270  NEXT I
3280 :
3290  REM  ................. analysis and display of data ..............
3300 :
3310  VTAB 13: POKE 1403,26: PRINT "Processing data ..."; CHR$ (7)
3320 :
3330  FOR I = 1 TO NUM
3332 Z(I) = Z(I) + Z(I + NUM) * 256
3340  IF Z(I) > MAX THEN MAX = Z(I)
3350  IF MIN > Z(I) THEN MIN = Z(I)
3360  NEXT I
3370 :
3380  GOSUB 13000: REM  show statistics on data
3390 :
3400  GOSUB 12000: REM  screen plot
3410 :
3420  GOSUB 16000: REM  Toggle hi-res pages
3430 :
3440  RETURN
3450 :
4000  REM  ------------------------------------------------------------
4010  REM  |                3.  Save Double Hires to binary            |
4020  REM  ------------------------------------------------------------
4030 :
4040  GOSUB 14000: REM  Get file name
4050 :
4060  &  SAVE FILE$,FILE$ + ".AUX"
4070 :
4080  RETURN
4090 :
5000  REM  ------------------------------------------------------------
5010  REM  |              4.  Save data array to text file             |
5020  REM  ------------------------------------------------------------
5030 :
5040  HOME
5050 :
5060  GOSUB 14000: REM  Get file name
5070 :
5080  PRINT D$;"OPEN";FILE$
5090  PRINT D$;"WRITE";FILE$
5100 :
5110  FOR I = 1 TO NUM
5120  PRINT Z(I)
5130  NEXT I
5140 :
5150  PRINT D$;"CLOSE";FILE$
5160 :
5170  RETURN
```

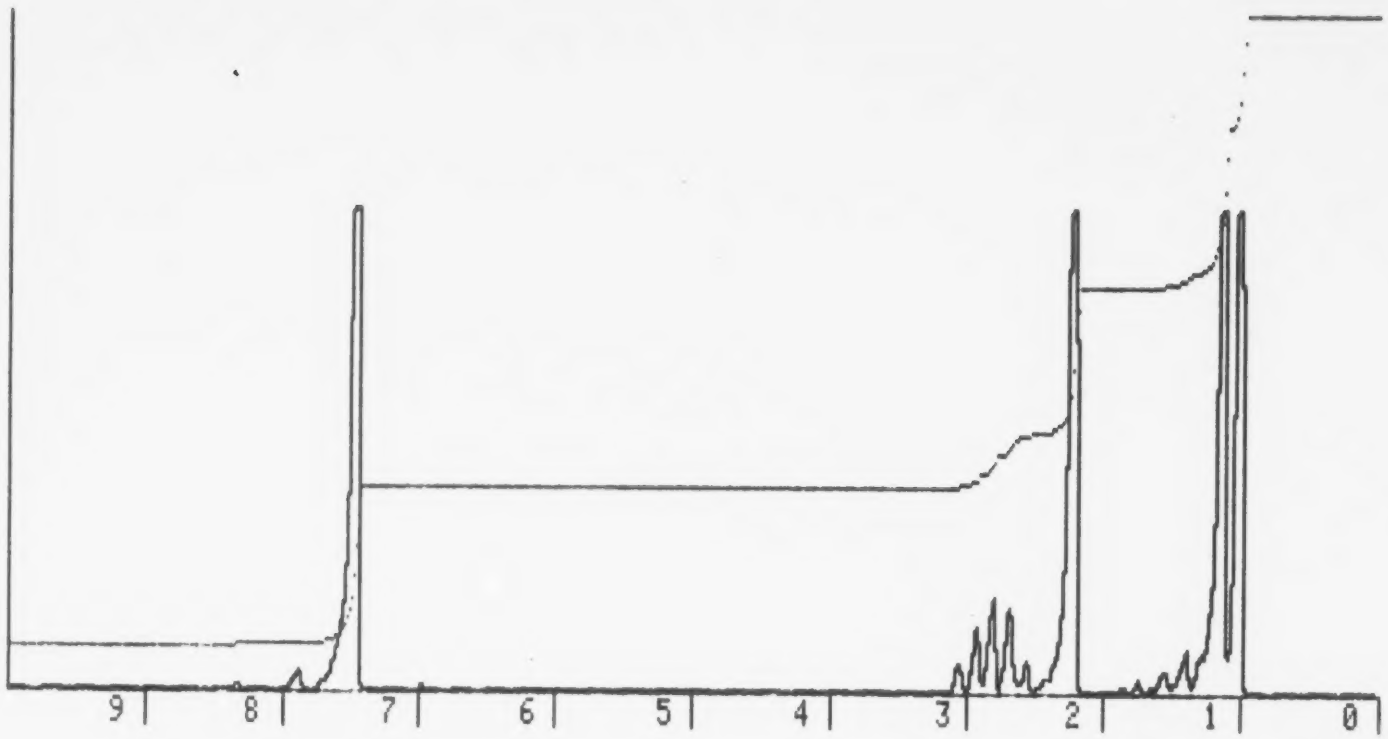
```
5180 :
6000  REM  ------------------------------------------------------------
6010  REM  !                    5.  Load graph from disk                !
6020  REM  ------------------------------------------------------------
6030 :
6040  HOME
6050 :
6060  GOSUB 14000: REM  Get file name
6070 :
6080  GOSUB 15000: REM  Viewing Instructions
6090 :
6100  &  LOAD FILE$,FILE$ + ".AUX"
6110 :
6120  GOSUB 9000: REM  Turn on Double Hi-res
6130 :
6140  GOSUB 16000: REM  Toggle hi-res pages
6150 :
6160  RETURN
6170 :
7000  REM  ------------------------------------------------------------
7010  REM  !                    6.  Load data from text file            !
7020  REM  ------------------------------------------------------------
7030 :
7040  HOME
7050 :
7060  GOSUB 14000: REM  Get file name
7065 :
7070  GOSUB 15000: REM  Viewing Instructions
7080 :
7090  PRINT D$;"OPEN";FILE$
7100  PRINT D$;"READ";FILE$
7110 :
7120  FOR I = 1 TO NUM
7130  INPUT Z(I)
7140  NEXT I
7150 :
7160  PRINT D$;"CLOSE";FILE$
7170 :
7180  GOSUB 12000: REM  Plot data
7190 :
7200  GOSUB 16000: REM  Toggle hi-res pages
7210 :
7220  RETURN
7230 :
8000  REM  ------------------------------------------------------------
8010  REM  !                       7.  Print routine                    !
8020  REM  ------------------------------------------------------------
8030 :
8040  HOME
8050 :
8060  VTAB 12: POKE 1403,32: PRINT "Make sure printer is on line"
8070  VTAB 14: POKE 1403,34: PRINT "Press any key to print ";: GET KEY$
8080 :
8090  PRINT  CHR$ (4): PRINT  CHR$ (4);"PR#1"
8100  PRINT  CHR$ (9);"G!DER"
8110  PRINT  CHR$ (4);"PR#3"
8120  PRINT
8130 :
8140  RETURN
8150 :
```

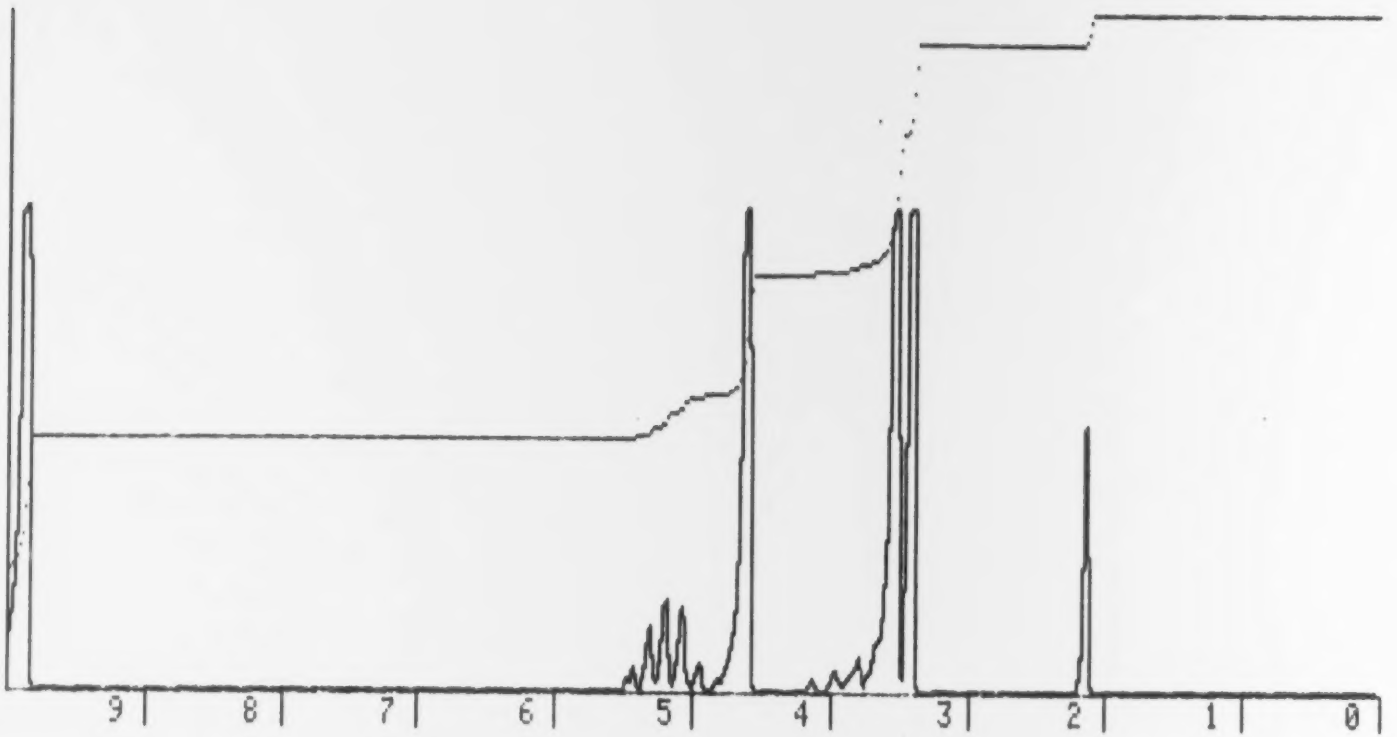
```
9000  REM  ------------------------------------------------------------
9010  REM  |                  8.  Turn on Double Hires                |
9020  REM  ------------------------------------------------------------
9030 :
9040  & MODE(1)
9050 :
9060  GOSUB 16000: REM  Toggle Hi-res pages
9070 :
9080  RETURN
9090 :
10000  REM  ------------------------------------------------------------
10010  REM  |                        Title Page                        |
10020  REM  ------------------------------------------------------------
10030 :
10040  PRINT  CHR$ (4)
10050  PRINT  CHR$ (4);"PR#3"
10060  PRINT
10070 :
10080  FOR I = 8 TO 74
10090  VTAB 01: POKE 1403,I: PRINT "_"
10100  VTAB 21: POKE 1403,I: PRINT "_"
10110  NEXT I
10120 :
10130  FOR I = 2 TO 21
10140  VTAB I: POKE 1403,06: PRINT "|"
10150  VTAB I: POKE 1403,75: PRINT "|"
10160  NEXT I
10170 :
10180  VTAB 04: POKE 1403,25: PRINT "     John Abbott College Cegep    "
10190  VTAB 07: POKE 1403,25: PRINT "Manoj Verghese and Beagle Brothers"
10200  VTAB 09: POKE 1403,25: PRINT "           present for            "
10210  VTAB 11: POKE 1403,25: PRINT "      The Chemistry Department    "
10220  VTAB 14: POKE 1403,25: PRINT "     Nuclear Magnetic Resonance   "
10230  VTAB 16: POKE 1403,25: PRINT "         Spectrum Display         "
10240  VTAB 18: POKE 1403,25: PRINT "In Double High Resolution Graphics"
10250 :
10260  FOR I = 1 TO 3000: NEXT I
10270 :
10280  RETURN
10290 :
11000  REM  ------------------------------------------------------------
11010  REM  |                       Display Menu                       |
11020  REM  ------------------------------------------------------------
11030 :
11040  TEXT : HOME
11050 :
11060  VTAB 01: POKE 1403,25: PRINT "    Men     "
11070  VTAB 04: POKE 1403,25: PRINT "Instructions ..... 1"
11080  VTAB 06: POKE 1403,25: PRINT "Run scan ......... 2"
11090  VTAB 08: POKE 1403,25: PRINT "Save spectrum ..... 3"
11100  VTAB 10: POKE 1403,25: PRINT "Save data ......... 4"
11110  VTAB 12: POKE 1403,25: PRINT "Load spectrum ..... 5"
11120  VTAB 14: POKE 1403,25: PRINT "Read data file ..... 6"
11130  VTAB 16: POKE 1403,25: PRINT "Dump spectrum ..... 7"
11140  VTAB 18: POKE 1403,25: PRINT "View current spectrum ..... 8"
11150  VTAB 20: POKE 1403,25: PRINT "Exit program ..... 9"
11160 :
11170  VTAB 24: POKE 1403,28: PRINT "      Enter selection : ";
11180  GET AN$
11190 :
```

```
11200  IF AN$ < "1" OR AN$ > "9" THEN 11170
11210 :
11220  RETURN
11230 :
12000  REM  ------------------------------------------------------------
12010  REM  !                          Screen Plot                       !
12020  REM  ------------------------------------------------------------
12030 :
12040  &  HGR2 : &  HCOLOR= 15
12050 :
12060  PRINT  CHR$ (4);"CATALOG,D1": REM  Change to drive 1"
12070  &  LOAD "AXIS.NMR.DHGR","AXIS.NMR.DHGR.AUX"
12080  PRINT
12090 :
12100 X = 0
12110 Y = 181 - Z(1) * 180 / 4095
12120  &  GOTO X,Y
12130 :
12140  FOR I = 2 TO NUM
12150 :
12160 X = X + 1
12170 Y = 181 - Z(I) * 180 / 4095
12180 :
12190  &  HPLOT  TO X,Y
12200 :
12210  NEXT I
12220 :
12230  GOSUB 17000: REM  Integration
12240 :
12250  RETURN
12260 :
13000  REM  ------------------------------------------------------------
13010  REM  !                     Shows stats on data                    !
13020  REM  ------------------------------------------------------------
13030 :
13040  HOME
13050 :
13060  VTAB 08: POKE 1403,22: PRINT "Minimum : ";MIN
13070  VTAB 10: POKE 1403,22: PRINT "Maximum : ";MAX
13080  VTAB 12: POKE 1403,22: PRINT "Range   : ";MAX - MIN + 1
13090 :
13100 MAX = 0
13110 MIN = 4096
13120 :
13130  VTAB 17: POKE 1403,25
13140  PRINT "Press any key to continue ";: GET KEY$
13150 :
13160  RETURN
13170 :
14000  REM  ------------------------------------------------------------
14010  REM  !                          File name                         !
14020  REM  ------------------------------------------------------------
14030 :
14040  HOME
14050 :
14060  PRINT D$
14070  PRINT D$;"CATALOG,D2"
14080  PRINT
14090  PRINT
14100 :
```

```
14110  INPUT "File name : ";FILE$
14120 :
14130  RETURN
14140 :
15000  REM  ------------------------------------------------------------
15010  REM  !                          Viewing Instructions                !
15020  REM  ------------------------------------------------------------
15030 :
15040  HOME
15050 :
15060  VTAB 04: POKE 1403,26: PRINT "Loading ";FILE$
15070  VTAB 08: POKE 1403,24: PRINT "Once you have viewed the graph"
15080  VTAB 10: POKE 1403,24: PRINT "pressing any key returns to the"
15090  VTAB 12: POKE 1403,24: PRINT "main menu.                      "
15100 :
15110  RETURN
15120 :
16000  REM  ------------------------------------------------------------
16010  REM  !                  Hi-Res Graphics Page Toggle                 !
16020  REM  ------------------------------------------------------------
16030 :
16040  GET KEY$: RETURN
16050 :
16060  REM  This program originally displayed two pages of hi-res.  It
16070  REM  was modified to display those two pages on a single page @
16080  REM  double hi-res ( Beagle Bros ).  In future if two pages of
16090  REM  double hi-res are to be displayed then a switch between
16100  REM  pages is needed which should be located here.  See Beagle
16110  REM  Bros 'Page.2' utility on Beagle Graphics disk.
16120 :
17000  REM  ------------------------------------------------------------
17010  REM  !                            Integration                       !
17020  REM  ------------------------------------------------------------
17030 :
17040  FOR I = 1 TO NUM
17050 Y = Y + Z(I)
17060  NEXT I: REM  Establish total area
17070 :
17080 FUDGE = Y / 170: REM  to keep integration curve on screen
17090 :
17100 X = 0
17110 Y = 0
17120 :
17130  FOR I = 1 TO NUM
17140 :
17150 X = X + 1
17160 Y = Y + Z(I)
17170 :
17180  &  HPLOT X,170 - Y / FUDGE
17190 :
17200  NEXT I
17210 :
17220  RETURN
17230 :
```


9
8
7
6
5
4
3
2
1
0


9
8
7
6
5
4
3
2
1
0


9
8
7
6
5
4
3
2
1
0

SPECTRUM

Spectrum

Introduction

"Spectrum" is a program which displays a spectrum on the graphics screen. Two methods can be used to obtain data for a plot. Data can be sampled from an analog to digital converter or a file containing data from a previous spectrum can be recalled. The program is currently set up for the Nmr.

Requirements

"Spectrum" will run on any IBM Pc series computer that will run Turbo Pascal 4.0 from Borland. The program was intended to run with a Hercules Monochrome card which provides a screen with 720x348 resolution. If a different card (such as a CGA) is used, minor adjustments should be made to account for the different resolution.

Three other files must be on the disk to run Spectrum. The "Herc.bgi" and "Graph.tpu" files are needed to enable the graphics mode. The "Atodutil" include file is needed to sample data from the analog to digital converter. For more information concerning the converter and the sampling program, please refer to "Voltage Sampling with a Computer - A Practical Project with an A/D Converter" by Joe Magee.

Main Program

The graphics screen is initialized. The title is displayed and the program waits for the user to press a key. A menu is presented with three options: sample through the A/D, recall a data file, or quit. If the first or second is chosen, a graph is drawn. When a key is pressed the menu is presented again. This procedure is repeated until the user decides to quit. Upon exiting the program, the display is restored to text mode.

Initialize

The appropriate graph driver is selected through the 'detect' function. (Different cards should not cause a problem.) The graphics screen is initialized.

Title

The screen is cleared and the text style is set to "triplex". The title is presented.

Select

The screen is cleared and the menu headed "Spectrum Source" is shown. The user is asked to enter his selection. If the response is invalid then the response defaults to number three which exits.

Pause

The program waits until the user presses a key. The key is read by the 'readkey' function.

Analog to Digital

The display is switched to text mode. The user is asked to press any key to start the sampling. The A/D is first initialized, then 719 samples taken which are displayed on the screen. Once finished the user is asked to supply a file name under which the data will be stored. No error checking is done on the name, so the user must take care to specify a valid filename.

Read File

The display is switched to text mode. The user is asked to supply the name of a data file. No error checking is done on the name. The file is opened and 719 points are read.

Axis

A set of labelled axes for an Nmr spectrum is drawn. The horizontal axis is marked from 10 to 0.

Integrate

An integration is performed by simply summing the data points. Care has been taken to ensure that the curve stays on the screen.

Adjust

This function scales the data to fit the graphics screen. When reading from files, a check may have to be performed to ensure proper scaling for the particular range of data.

Draw

The spectrum is drawn using the 'moveto' and 'lineto' procedures.

```
program spectrum;

uses Crt, Graph;

{$R-}    {Range checking off}
{$B+}    {Boolean complete evaluation on}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$M 65500,16384,655360} {Turbo 3 default stack and heap}
ERROR:  include file not found ($i)

const
     n  =  719;     { 720 = screen resolution, 558 = nmr data, 1116 = ir data

var
     a              :  array [1..n] of integer;
     graph_driver   :  integer;
     graph_mode     :  integer;
     choice         :  char;

{ ........................... Definition of Variables .....................

     a              :  array of spectrum points taken from atod
     graph_driver   :  HERC
     graph_mode     :  HercMonoHi
     choice         :  A - atod     B - data file

  ............................ Dictionary of Procedures .....................

     Initialize  :  intializes the graphics screen
     Title       :  displays the program title
     Select      :  get input from an AtoD or a datafile
     Pause       :  waits for the user to press any key to continue
     Atod        :  reads an analog to digital converter
     Read_file   :  reads a data file
     Integrate   :  integration routine for Nmr graph
     Draw        :  draws the corresponding graph from the data collected

{ ------------------------------------------------------------------------
{ :                                Initialize                            :
{ ------------------------------------------------------------------------

procedure initialize;

begin

     graph_driver := detect;
     initgraph ( graph_driver, graph_mode, '' );

end;  { of initialize }

{ ------------------------------------------------------------------------
{ :                                Title                                 :
{ ------------------------------------------------------------------------
```

```
procedure title;

begin

     cleardevice;
     settextstyle ( triplexfont, horizdir, 4 );

     rectangle    ( 0, 0, 719, 347 );
     rectangle    ( 5, 3, 714, 344 );

     outtextxy    ( 220, 80, 'Analog to Digital' );
     outtextxy    ( 220,140, '   Conversion    ' );
     outtextxy    ( 210,200, 'Graphical Display' );

end;  { of title }

{ ------------------------------------------------------------------------------ }
{ :                                Select                                      : }
{ ------------------------------------------------------------------------------ }

procedure select;

begin

     cleardevice;
     settextstyle ( triplexfont, horizdir, 3 );

     rectangle ( 160, 20, 555, 70 );
     rectangle ( 157, 17, 558, 73 );

     outtextxy ( 215, 30, '   Spectrum Source   ' );
     outtextxy ( 210,110, 'Analog to Digital . . . . 1' );
     outtextxy ( 210,150, 'Data file . . . . . . . . 2' );
     outtextxy ( 210,190, 'Exit . . . . . . . . . . . 3' );
     outtextxy ( 220,260, '  Enter selection :  ' );

     readln ( choice );
     if ( choice <> '1' ) and ( choice <> '2' ) then choice := '3';
     outtextxy ( 450, 260, choice );

     delay ( 2500 );

end;  { of select }

{ ------------------------------------------------------------------------------ }
{ :                                Pause                                       : }
{ ------------------------------------------------------------------------------ }

procedure pause;

var
     character  :  char;

begin
```

```
     repeat until keypressed;
     character := readkey;

end;  { of pause }

{ ------------------------------------------------------------------------------ }
{ :                          Analog to Digital                                 : }
{ ------------------------------------------------------------------------------ }

procedure atod;

var
     i,j   :  integer;
     temp  :  longint;
     name  :  string[11];
     list  :  file of integer;

begin

     restorecrtmode;
     clrscr;

     writeln('                              Press any key to start sampling');
     writeln;
     pause;

          atodinit;

               for i := 1 to n do
               begin

                    temp  := 0;
                    for j := 1 to 410 do temp := temp + read_atod;
                    a[i]  := temp div 410;

                    write ( a[i]:5 );

               end;

     writeln;
     writeln;
     gotoxy ( 20, 25 );
     write ('Enter the target file name : ');
     readln ( name );

          assign  ( list, name );
          rewrite ( list );
               for i := 1 to n do write ( list, a[i] );
          close ( list );

end;  { of atod }

{ ------------------------------------------------------------------------------ }
{ :                              Read File                                     : }
{ ------------------------------------------------------------------------------ }
```

```
procedure read_file;

var
     i     :  integer;
     name  :  string[11];
     list  :  file of integer;

begin

     restorecrtmode;
     clrscr;

     gotoxy ( 20,10 );
     write('Enter the source file name : ');
     readln ( name );

          assign ( list, name );
          reset  ( list );
               for i := 1 to n do read ( list, a[i] );
          close ( list );

end;  { of read_file }

{ ------------------------------------------------------------------------ }
{ :                          Labelled Axis                               : }
{ ------------------------------------------------------------------------ }

procedure axis;

var
     i  :  integer;
     s  :  string[1];

begin

     line ( 0,   0,   0, 335 );                          { vertical axis }
     line ( 0, 335, 719, 335 );                        { horizontal axis }

     for i := 0 to 9 do line ( i*72, 335 , i*72, 347 );  { 10 to 1 markings }

     line ( 719, 335, 719, 347 );                          { 0 marking }

     for i := 9 downto 0 do
     begin

          str ( i, s );
          outtextxy ( 700 - i*72 , 340, s );

     end;

end;  { of axis }

{ ------------------------------------------------------------------------ }
{ :                       Integration Routine                            : }
```

```
{ ------------------------------------------------------------------------------

procedure integrate;

var
     i, y, fudge, point:  integer;

begin

     y := 0;

     for i := 1 to n do      y := y + a[i];
                          fudge := y div 310;
                              y := 0;

     for i := 1 to n do
     begin

          y := y + a[i];
          point := y div fudge;
          putpixel ( i, 320 - point, 1 );

     end;

end;  { of integrate }

{ ------------------------------------------------------------------------------
{ :                                  Adjust                                    :
{ ------------------------------------------------------------------------------

{         The dividing factor will have to be changed to suit the range of
     data that is being used.  This usually depends on the AtoD: in this
     case, it is 256 ( 8 bit AtoD ).

function  f ( a : integer )  : integer;

begin

     f := 334 - round ( 333.0 * a /  255.0 );

end;  { of adjust }

{ ------------------------------------------------------------------------------ }
{ :                                  Draw                                      : }
{ ------------------------------------------------------------------------------ }

procedure draw;

var
     x  :  integer;

begin

     setgraphmode ( graph_mode );
     cleardevice;
```

```
      axis;                                              { draw labelled axis

           x := 1;
           moveto ( 1, f(a[x]) );                        { set starting position

      for x := 2 to n do
      begin
           lineto ( x, f(a[x]) );
           moveto ( x, f(a[x]) );
      end;

      integrate;                                         { integrate Nmr graph

end;  { of draw }

{ ------------------------------------------------------------------------- }
{ :                              Main Block                                 : }
{ ------------------------------------------------------------------------- }

begin

      initialize;                                   { initialize graphics screen

      title;                                             { display program title
      pause;                                                  { wait for user

           repeat

                select;                                   { select input source

                case choice of
                     '1'   :  atod;                            { sample data or
                     '2'   :  read_file;                       { read data file
                end;

                if choice <> '3' then
                begin
                     draw;                                     { draw the graph
                     pause;                                     { wait for user
                end;

           until ( choice = '3' );

      closegraph;                                        { return to text mode

end.  { of main block }
```

REALTIME

Realtime

Introduction

"Realtime" is a program which draws a graph on the graphics screen in real time. Data sampled from an analog to digital converter is plotted immediately, not after having done a full scan.

Requirements

"Realtime" will run on any IBM Pc series computer that will run Turbo Pascal 4.0 from Borland. The program was intended to run with a Hercules Monochrome card which provides a screen with 720x348 resolution. If a different card (such as a CGA) is used then minor adjustments should be made to account for the different resolution.

Three other files must be on the disk to run Realtime. The "Herc.bgi" and "Graph.tpu" files are needed to enable the graphics mode. The "Atodutil" include file is needed to sample data from the analog to digital converter. For more information concerning the converter and the sampling program, please refer to "Voltage Sampling with a Computer - A Practical Project with an A/D Converter" by Joe Magee.

Main Block

The graphics screen is initialized and so is the A/D. The title is displayed and after a brief pause the plotting begins. If the voltage is varied, the effect can be seen immediately on the screen. Once a full screen has been drawn, the program waits for a key to be pressed. If 'q' is pressed, the program quits, any other key causes another screen to be plotted. Upon exiting the program, the display is restored to text mode.

Initialize

The appropriate graph driver is selected through the 'detect' function. (Different cards should not cause a problem.) The graphics screen is initialized.

Title

The screen is cleared and the text style is set to "triplex". The title is presented. The program continues after a brief pause.

Pause

The program waits until the user presses a key. The key is read by the 'readkey' function.

Real_time_plot

The graphics screen is cleared and sampling begins. Plotting is accomplished through the use of 'moveto' and 'lineto' procedures.

```
program real_time;

uses Crt, Graph;

{$R-}    {Range checking off}
{$B+}    {Boolean complete evaluation on}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$M 65500,16384,655360} {Turbo 3 default stack and heap}
ERROR:  include file not found ($i)

const
     n  =  719;    { 720 = screen resolution, 558 = nmr data, 1116 = ir data }

var
     character  :  char;

{ ........................... Definition of Variables .....................

     pixel          :  spectrum points taken from atod
     graph_driver   :  HERC
     graph_mode     :  HercMonoHi
     i,j,k          :  loop counters
     character      :  user input char

   ........................... Dictionary of Procedures .....................

     Initialize       :  intializes the graphics screen
     Title            :  displays the program title
     Pause            :  waits for the user to press any key to continue
     Real_time_plot   :  reads an analog to digital converter

{ ------------------------------------------------------------------------- }
{ :                                Initialize                             : }
{ ------------------------------------------------------------------------- }

procedure initialize;

var
     graph_driver, graph_mode : integer;

begin

     graph_driver := detect;
     initgraph ( graph_driver, graph_mode, '' );

end;  { of initialize }

{ ------------------------------------------------------------------------- }
{ :                                  Title                                : }
{ ------------------------------------------------------------------------- }

procedure title;

begin
```

```
     cleardevice;
     settextstyle ( triplexfont, horizdir, 4 );

     rectangle    ( 0, 0, 719, 347 );
     rectangle    ( 5, 3, 714, 344 );

     outtextxy    ( 210, 80, 'Real Time Plotting' );
     outtextxy    ( 210,140, '         by         ' );
     outtextxy    ( 210,200, '  Manoj Verghese   ' );

     delay ( 1500 );

end;  { of title }

{ ------------------------------------------------------------------------------
{ :                                  Pause                                     :
{ ------------------------------------------------------------------------------

procedure pause;

begin

     repeat until keypressed;
     character := readkey;

end;  { of pause }

{ ------------------------------------------------------------------------------
{ :                              Real Time Plot                                :
{ ------------------------------------------------------------------------------

procedure real_time_plot;

var
     i,j,k  :  integer;
     pixel  : longint;

begin

     cleardevice;

          for i := 1 to n do
          begin

               for k := 1 to 100 do
               begin
                    pixel := pixel + read_atod;
               end;

                    pixel := pixel div 100;
                        j := 347 - round ( 344.0 * pixel / 255.0 );

               lineto ( i, j );
               moveto ( i, j );
```

```
          end;

end;  { of draw }

{ ------------------------------------------------------------------------ }
{ |                              Main Block                              | }
{ ------------------------------------------------------------------------ }

begin

     initialize;                                  { initialize graphics screen
     atodinit;                                              { initialize AtoD

     title;                                          { display program title

          repeat

               real_time_plot;                                { plot the graph
               pause;                                          { wait for user

          until ( character = 'q' );                             { 'q' to quit

     restorecrtmode;
     closegraph;                                    { return to text mode

end.  { of main block }
```

DIFFICULTIES

Difficulties

Graphics Writer

Applesoft provides the programmer with memory starting from $800 up to $9600. In theory, this means that any basic program with its variables can be over 30 000 bytes long. However, hi-res page one occupies $2000 to $3FFF and page two occupies $4000 to $5FFF. This is right in the middle of memory. If your program and its variables do not fit between $800 and $2000 and uses graphics there will be a conflict in memory. The following two possibilities exist. If after loading or when running a program, you issue an Hgr (Hgr2) command, part of your program and/or variables will be deleted. If you issue an Hgr (Hgr2) command first and then load or run, little lines will show up on the hi-res pages.

There are a few different solutions to the problem:

If the program uses only a single page of graphics, then use page two instead of page one. This provides an extra $2000 (8192) bytes.

If only the variables (which are stored just above the program) are the problem then a Lomem : 16383 or 24576 command can be issued at the beginning of the program telling Applesoft to store variable information above page one or two.

If the program goes into page two or uses both pages and the above solutions do not work then the program has to be loaded above page two. This involves poking the program pointers with a new location in memory.

This is done in the case of "Graphics Writer". A routine from Beagle Brothers is used to accomplish this. The program resides from $6000 to $9043. The two font files reside from $4000 to $46F6 Page one is used for graphics pictures.

There is one other problem with "Graphics Writer". It seems that if the two font files, "Standard" and "Cursors" are not loaded before "Graphics Writer" is run, they will not work. I have not discovered why. This is the reason for the "Startup" program which bloads the fonts and then runs "Graphics Writer".

Also, it should be noted that the ascii value for the right arrow key changes according to whether the 80 column card is on or off. On, the value is chr$(32) (also the spacebar), off

chr$(21). The program runs with the 80 column card on and in 40 column mode. 40 column mode is chosen because the pokes which control the graphics do not work in 80 column mode.

Ir.Plot

The "Ir.Plot" program is $2249 bytes in length. Since this would occupy part of hi-res page one, page two is used to display the spectrum. For a more thorough discussion of Apple's memory map, please refer to the section under "Graphics Writer".

The only drawback to this program is speed. The number of points sampled is 1116. However, only a quarter (279) of these points are used in the spectrum. This is a limitation of Apple's resolution. Since the extra points are being stored, disk access time is less than optimal. The additional data was collected so that in future, another computer with a higher resolution could display the spectra.

Nmr.Plot

The "Nmr.Plot" program is $28A3 bytes in length. Since the program and its variables would occupy most of page one and both pages are used, the program is moved above page two. For a more detailed discussion of Apple's memory map, please refer to the section under "Graphics Writer".

Synchronizing the sampling time with the scan time of the Nmr caused some difficulty. The 12 bit A/D returns values from 0 to 4095, in other words a 2 byte number. Conversion to a decimal value is done by multiplying the most significant byte by 256 (2^8) and adding it to the least significant byte. The time taken for multiplication varies according to the number being converted. Thus the sampling time varies by a few seconds for different compounds. The problem was resolved by storing the Msb and Lsb during the sampling and leaving the conversion till later.

DISK CATALOG

Graphics Utilities

Challenge '88

*A 003 HELLO
*A 054 GRAPHICS WRITER
*A 003 STARTUP
*B 034 TITLE
*B 003 CURSORS
*B 007 STANDARD

Ir - sampling & display programs

Challenge '88

*A 003 HELLO
*B 034 AXIS.IR
*A 010 IR.BI
*A 008 IR.QUAD
*A 037 IR.PLOT

Ir - binary & text files

Challenge '88

*B 034 ANILINE.BIN
*T 023 ANILINE.DTA
*B 034 BENZOIC ACID.BIN
*T 023 BENZOIC ACID.DTA
*B 034 BUTANOL-1.BIN
*T 023 BUTANOL-1.DTA
*B 034 DIETHYLAMINE.BIN
*T 023 DIETHYLAMINE.DTA
*B 034 HEPTANOL.BIN
*T 023 HEPTANOL.DTA
*B 034 HYDROCYNNAMIC ACID.BIN
*T 023 HYDROCYNNAMIC ACID.DTA
*B 034 PHENOL.BIN
*T 023 PHENOL.DTA
*B 034 PROPANOL.BIN
*T 023 PROPANOL.DTA
*B 034 XYLENE.BIN
*T 023 XYLENE.DTA

```
-------------------------------------------------
    Nmr - sampling & display programs * hi-res *
-------------------------------------------------

Challenge '88

*A 002 BOOT.SYSTEM
*A 003 HELLO
*B 066 AXIS.NMR
*B 006 DOS MOVER
*A 044 NMR.PLOT

-------------------------------------------------
        Nmr - binary & text files * hi-res *
-------------------------------------------------


Challenge '88

*B 066 NMR1.BIN
*T 010 NMR1.DTA
*B 066 NMR2.BIN
*T 007 NMR2.DTA
*B 066 NMR3.BIN
*T 007 NMR3.DTA

-------------------------------------------------
Nmr - sampling & display programs * double hi-res *
-------------------------------------------------


Challenge '88

*A 002 BOOT.SYSTEM
*A 003 HELLO
*B 034 AXIS.NMR.DHGR
*B 034 AXIS.NMR.DHGR.AUX
*B 020 DHGR
*B 006 DOS MOVER
*A 043 NMR.PLOT.DHGR

-------------------------------------------------
    Nmr - binary & text files * double hi-res *
-------------------------------------------------


Challenge '88

*B 034 NMR1
*B 034 NMR1.AUX
*T 010 NMR1.DTA
*B 034 NMR2
*B 034 NMR2.AUX
*T 007 NMR2.DTA
*B 034 NMR3
*B 034 NMR3.AUX
*T 007 NMR3.DTA
```

```
       24 File(s)     858112 bytes free
%PIPE1    $$$          0   8-12-88    9:58a
%PIPE2    $$$          0   8-12-88    9:58a
HERC      BGI       5933  11-17-87    4:00a
LITT      CHR       2126  11-17-87    4:00a
SANS      CHR       5452  11-17-87    4:00a
TRIP      CHR       7213  11-17-87    4:00a
GOTH      CHR       8489  11-17-87    4:00a
COMMAND   COM      23210   3-07-85    1:43p
NMR1      DTA       1116   7-28-88   11:37a
NMR2      DTA       1116   7-28-88   11:39a
NMR3      DTA       1116   7-28-88   11:39a
SAMPLE1   DTA       1438   8-09-88   11:29a
SAMPLE2   DTA       1438   8-09-88   11:35a
ANILINE   DTA       2232   7-28-88   11:32a
BENZOIC   DTA       2232   7-28-88   11:35a
PHENOL    DTA       2232   7-28-88   11:36a
ATOD      EXE      10560   8-12-88    9:48a
REALTIME  EXE      20464   7-29-88   11:48a
SPECTRUM  EXE      22944   8-09-88    2:06p
 Volume in drive A is TURBOPASCAL
ATOD      PAS       1085   7-27-88   11:27a
ATODUTIL  PAS       3337   7-28-88    8:33a
REALTIME  PAS       4043   8-05-88    8:22a
SPECTRUM  PAS       9341   8-10-88    8:47a
GRAPH     TPU      27184  11-17-87    4:00a
 Directory of  A:\
```